



CFO 15888 US / sei

日本特許庁  
JAPAN PATENT OFFICE

09/978,214

Tadashi Yamakawa

October 17, 2001

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2001年 4月27日

出願番号

Application Number:

特願2001-132795

出願人

Applicant(s):

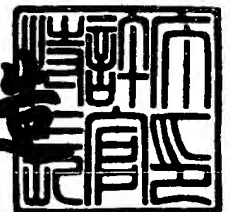
キヤノン株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年11月 9日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3098754

【書類名】 特許願

【整理番号】 4278154

【提出日】 平成13年 4月27日

【あて先】 特許庁長官殿

【国際特許分類】 H04L 12/00

【発明の名称】 遠隔操作システム、遠隔操作方法、被操作装置、操作端末、プログラムおよび記憶媒体

【請求項の数】 30

【発明者】

    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

    【氏名】 中川 久雄

【発明者】

    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

    【氏名】 亀井 洋一

【発明者】

    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

    【氏名】 山川 正

【特許出願人】

    【識別番号】 000001007

    【氏名又は名称】 キヤノン株式会社

    【代表者】 御手洗 富士夫

【代理人】

    【識別番号】 100081880

    【弁理士】

    【氏名又は名称】 渡部 敏彦

    【電話番号】 03(3580)8464

【手数料の表示】

【予納台帳番号】 007065

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9703713

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 遠隔操作システム、遠隔操作方法、被操作装置、操作端末、プログラムおよび記憶媒体

【特許請求の範囲】

【請求項 1】 ネットワークを介して被操作装置および操作端末が接続され、前記操作端末で作成された操作情報を用いて前記被操作装置を操作する遠隔操作システムであって、

前記被操作装置は、前記操作端末への通信を発動し、前記操作情報を前記操作端末に要求する操作情報要求手段を備え、

前記操作端末は、前記発動された通信に応じ、所定時間だけ操作の発生を待機した後、前記要求された操作情報を前記被操作装置に返す操作情報応答手段を備えたことを特徴とする遠隔操作システム。

【請求項 2】 前記被操作装置は、前記操作情報の要求とは別に、前記操作端末への通信を発動し、該被操作装置の状態情報を前記操作端末に通知する状態情報通知手段を備え、

前記操作端末は、前記発動された通信に応じ、待機することなく、前記通知された状態情報の応答を前記被操作装置に返す状態情報通知応答手段を備えたことを特徴とする請求項 1 記載の遠隔操作システム。

【請求項 3】 前記操作情報要求手段は、前記操作端末への通信を発動して前記被操作装置の状態情報を通知するとともに、該通信を用いてその応答としての前記操作情報を要求し、

前記操作情報応答手段は、前記通知された状態情報の応答として、前記通信を用いて前記要求された操作情報を返すことを特徴とする請求項 1 記載の遠隔操作システム。

【請求項 4】 前記被操作装置は、前記操作端末との通信を制御する通信制御手段を独立に備え、

前記操作情報要求手段は、前記通信制御手段を用いて前記操作端末への通信を発動することを特徴とする請求項 1 または 2 記載の遠隔操作システム。

【請求項 5】 前記被操作装置および前記操作端末間の通信プロトコルとし

て、HTTPプロトコルを使用することを特徴とする請求項1記載の遠隔操作システム。

【請求項6】 前記操作端末は、HTTPサーバプログラムを用いて、前記被操作装置からの情報を受信し、前記状態情報通知応答手段および前記操作情報応答手段を、前記HTTPサーバプログラムから起動されるCGIプログラムによって実現することを特徴とする請求項5記載の遠隔操作システム。

【請求項7】 前記操作情報応答手段は、前記所定時間内に操作が発生しなかった場合、操作なしである旨の前記操作情報を作成することを特徴とする請求項1記載の遠隔操作システム。

【請求項8】 前記所定時間は、前記被操作装置および前記操作端末間のネットワークあるいは通信プロトコルによって規定される、データ送受信がない期間が持続した際のコネクション切断タイムアウト時間より短い時間に設定されることを特徴とする請求項7記載の遠隔操作システム。

【請求項9】 前記状態情報通知手段、前記操作情報要求手段、前記状態情報通知応答手段および前記操作情報応答手段は、複数の通信を用いて並列に動作することを特徴とする請求項2記載の遠隔操作システム。

【請求項10】 前記操作情報要求手段または前記操作情報応答手段が並列に動作している場合、前記操作が発生するまで待機している前記操作情報応答手段の数を所定数に制限し、該所定数を越える分の前記操作情報応答手段は、前記操作なしである旨の前記操作情報を作成することを特徴とする請求項7記載の遠隔操作システム。

【請求項11】 前記被操作装置および前記操作端末間の通信コネクションとして、HTTP規定の持続的コネクションを使用することを特徴とする請求項5記載の遠隔操作システム。

【請求項12】 HTTP規定のパイプライン処理を使用することを特徴とする請求項11記載の遠隔操作システム。

【請求項13】 データ送受信がない期間が所定時間以上継続した場合、データ送受信を行い、前記持続的コネクションを継続させることを特徴とする請求項11または12記載の遠隔操作システム。

【請求項 1 4】 操作端末で作成された操作情報を用い、ネットワークを介して被操作装置を操作する遠隔操作方法であって、

前記被操作装置から前記操作端末へ通信を発動し、前記操作情報を前記操作端末に要求する操作情報要求工程と、

前記発動された通信に応じ、所定時間だけ操作の発生を待機した後、前記操作端末から前記要求された操作情報を前記被操作装置に返す操作情報応答工程とを有することを特徴とする遠隔操作方法。

【請求項 1 5】 前記操作情報の要求とは別に、前記操作端末への通信を発動し、前記被操作装置の状態情報を前記操作端末に通知する状態情報通知工程と

前記発動された通信に応じ、待機することなく、前記通知された状態情報の応答を前記被操作装置に返す状態情報通知応答工程とを有することを特徴とする請求項 1 4 記載の遠隔操作方法。

【請求項 1 6】 前記操作情報要求工程では、前記操作端末への通信を発動して前記被操作装置の状態情報を通知するとともに、該通信を用いてその応答としての前記操作情報を要求し、

前記操作情報応答工程では、前記通知された状態情報の応答として、前記通信を用いて前記要求された操作情報を返すことを特徴とする請求項 1 4 記載の遠隔操作方法。

【請求項 1 7】 前記操作情報要求工程では、独立に備えられた通信制御手段を用いて前記操作端末への通信を発動することを特徴とする請求項 1 4 または 1 5 記載の遠隔操作方法。

【請求項 1 8】 前記被操作装置および前記操作端末間の通信プロトコルとして、HTTP プロトコルを使用することを特徴とする請求項 1 4 記載の遠隔操作方法。

【請求項 1 9】 前記操作端末は、HTTP サーバプログラムを用いて、前記被操作装置からの情報を受信し、前記状態情報通知応答工程および前記操作情報応答工程は、前記 HTTP サーバプログラムから起動される CGI プログラムによって実現されることを特徴とする請求項 1 8 記載の遠隔操作方法。

【請求項 2 0】 前記操作情報応答工程では、前記所定時間内に操作が発生しなかった場合、操作なしである旨の前記操作情報を作成することを特徴とする請求項 1 4 記載の遠隔操作方法。

【請求項 2 1】 前記所定時間は、前記被操作装置および前記操作端末間のネットワークあるいは通信プロトコルによって規定される、データ送受信がない期間が持続した際のコネクション切断タイムアウト時間より短い時間に設定されることを特徴とする請求項 2 0 記載の遠隔操作方法。

【請求項 2 2】 前記状態情報通知工程、前記操作情報要求工程、前記状態情報通知応答工程および前記操作情報応答工程は、複数の通信を用いて並列に行われることを特徴とする請求項 1 5 記載の遠隔操作方法。

【請求項 2 3】 前記操作情報要求工程または前記操作情報応答工程が並列に行われる場合、前記操作が発生するまで待機している前記操作情報応答工程の数を所定数に制限し、該所定数を越える分の前記操作情報応答工程では、前記操作なしである旨の前記操作情報を作成することを特徴とする請求項 2 0 記載の遠隔操作方法。

【請求項 2 4】 前記被操作装置および前記操作端末間の通信コネクションとして、H T T P 規定の持続的コネクションを使用することを特徴とする請求項 1 8 記載の遠隔操作方法。

【請求項 2 5】 H T T P 規定のパイプライン処理を使用することを特徴とする請求項 2 4 記載の遠隔操作方法。

【請求項 2 6】 データ送受信がない期間が所定時間以上継続した場合、データ送受信を行い、前記持続的コネクションを継続させることを特徴とする請求項 2 4 または 2 5 記載の遠隔操作方法。

【請求項 2 7】 ネットワークに接続された操作端末から操作される被操作装置であって、

前記操作端末への通信を発動し、状態情報を通知する状態情報通知手段と、

前記操作端末への通信を発動し、操作情報を要求する操作情報要求手段とを備え、

前記操作端末は、前記被操作装置からの通信の発動に応じ、前記通知された状

態情報の応答を待機することなく返す一方、前記要求された操作情報を所定時間待機した後に返すことを特徴とする被操作装置。

【請求項 28】 ネットワークに接続された被操作装置を操作する操作端末であって、

前記被操作装置からの通信の発動に応じ、前記被操作装置から通知された状態情報の応答を待機することなく前記被操作装置に返す状態情報通知応答手段と、

前記被操作装置からの通信コネクションの発動に応じ、前記被操作装置から要求された操作情報を、所定時間待機した後に前記被操作装置に返す操作情報応答手段とを備えたことを特徴とする操作端末。

【請求項 29】 請求項 14 乃至 26 のいずれかに記載の遠隔操作方法を実現するためのプログラムコードを保持する記憶媒体。

【請求項 30】 請求項 14 乃至 26 のいずれかに記載の遠隔操作方法を実現するためのプログラムコードを有するプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、操作端末を用いて遠隔から機器を操作する遠隔操作システム、遠隔操作方法、被操作装置、操作端末、プログラムおよび記憶媒体に関する。

【0002】

【従来の技術】

従来、ネットワークを介して機器と操作端末を結ぶ遠隔操作システムが知られている。このような遠隔操作システムでは、LAN やインターネットの普及と共に、計算機やネットワーク機器だけでなく、種々の機器をネットワークに接続し、ネットワークを介して情報を取得したり、制御することが普及しつつある。例えば、遠隔地の温度計を読み取ったり、ビデオカメラの雲台を制御することが挙げられる。

【0003】

このような遠隔操作システムは、操作対象の機器と操作する機器との双方がネットワークに接続され、所定の通信プロトコルで通信することによって実現され



る。

【0004】

しかし、操作対象機器と操作用機器とが通信できない場合がある。多くの組織が自組織のLANをインターネットに接続する場合、ファイアウォールと称される通信の制限手段をその接続点に設けることが行われている。この制限手段は、LANをインターネット上の悪意の利用者から守るためであるが、同時にLAN上の正当な利用者のインターネット利用も妨げてしまう場合もある。

【0005】

例えば、ファイアウォールの実現手段の1つとして、LANとインターネットとの間で特定の条件、例えば、通信プロトコル、送信元、送信先を満たした通信パケットだけを通過させるように制限することが広く行われている。

【0006】

また、サーバ/クライアントモデルとストリーム型通信プロトコル(TCP/IP)の特性に基づき、LAN内のクライアントが通信要求を発し、インターネット上のサーバを利用できるようにしているが、逆にLAN上のサーバに対してインターネットから通信要求を伝えることができないようにすることが一般的である。

【0007】

したがって、LANの一般利用者は、インターネットからの通信要求を受けるサーバをLAN内に置くことは、通常できない。

【0008】

さらに、ファイアウォールでは、LANとインターネットとの間で通信パケットをOSレベルで(通信プロトコルスタックで)中継しないように制限することも行われる。この場合、安全性は高まるが、インターネットからLAN方向だけでなく、LANからインターネットへ出ていく通信パケットもファイアウォールを通過できなくなる。

【0009】

そこで、アプリケーションレベルのプログラムで通信パケット全体ではなく通信すべきデータのみを中継するようにし、必要なデータの通信を実現することが

行われている。代表的な中継プログラムとして、WWW (World Wide Web) のデータを中継するプロキシサーバが知られており、多くのファイアウォール上でサービスが提供されている。しかし、全てのアプリケーションプロトコルに対して中継プログラムが用意されているわけではない。したがって、LANの一般利用者は全てのアプリケーションプロトコルを利用できるとは限らない。

#### 【0010】

このように、ファイアウォールが機器間に存在する場合、操作対象機器と操作機器との間の通信は、一般に制約されることが多い。機器のネットワークの接続先が制限されたり、遠隔操作の通信プロトコルがファイアウォールを通過できなかったりすることがある。これに対し、ファイアウォールの設定を変えることで制約を取り除くことは可能であるが、ファイアウォールの安全性を弱めるので、一般的に認められないことが多い。

#### 【0011】

また、ファイアウォール管理者への依頼事務手続きが煩雑であったり、管理者が多忙で頻繁な変更には応じられないという場合も多く、ファイアウォールの設定を変えることは避けられている。

#### 【0012】

そこで、独自の通信プロトコルを使用するのではなく、ファイアウォールを通過できるように設定されていることの多い、WWWの通信プロトコルであるHTTP (Hyper Text Transfer Protocol) を用いる方法が提案されている。即ち、独自のプロトコルをHTTPのメソッドやデータに変換したり、独自のプロトコルを埋め込んでHTTPとしてファイアウォールを通過させ、独自プロトコルのコマンドやデータに再構成する。これは、機器操作に限らず、音声や動画等の連続データのファイアウォール通過にも広く用いられている。

#### 【0013】

ところで、HTTPは、前述したストリーム型通信プロトコル(TCP/IP)上にサーバ/クライアントモデルのアプリケーションプロトコルを構築したも

のであるが、クライアントからのリクエストとサーバからのレスポンスの1往復の通信が基本となっている。

## 【0014】

当初のプロトコルでは、この1往復の通信を1つのストリーム型通信コネクションに割り当てており、1つのリクエスト毎にストリーム型コネクションを確立する必要があり、そのオーバーヘッドが大きかった。

## 【0015】

そこで、現在のプロトコルでは、持続的コネクション (persistent connection) が規定され、1つの通信コネクションを複数往復分のリクエスト・レスポンス交換に割り当てられるようになっている。さらに、レスポンスが返る前に続けてリクエストを送るパイプライン処理も規定されている。

## 【0016】

## 【発明が解決しようとする課題】

このように、HTTPの通信の枠組みを利用することで、ファイアウォール内部のLANに接続された操作用機器からインターネット上の操作対象機器への通信・制御は可能になる。

## 【0017】

しかしながら、逆方向、つまりインターネット上の操作用機器からLANに接続された操作対象機器への通信・制御のための通信要求がファイアウォールを通過できないという問題が依然として残っていた。

## 【0018】

これに対し、内部の機器から外部の機器に向けて、一定間隔で継続して接続要求を行い、外部の機器は接続を行う時だけ、これに応答する、いわゆるポーリング方式を利用することが考えられる。ポーリング方式は、例えば、ファクシミリ情報サービスにFAXメッセージ受信を要求するといった場合に利用される。

## 【0019】

しかし、遠隔操作のような場合、FAX受信の場合と違い、即時応答性が重要となる。また、外部からの操作は不定期であり、内部の機器側では、外部からの操作がいつ行われるかを予測できないことに加え、複数の操作要求が行われるこ

とが予想されるので、頻繁に接続要求を行う必要があった。

【 0 0 2 0 】

したがって、ポーリング方式を用い、遠隔操作のような即時応答性を重要視した通信を行う際、常に通信トラフィックが発生し、かつそのうちの殆どが無駄なトラフィックとなり得るという問題を抱えていた。

【 0 0 2 1 】

さらに、この問題を解決する過程で、持続的コネクションを用いて通信コネクション確立までの遅延時間を減少させることが考えられるが、機器間の HTTP データを中継するプロキシサーバで設定される持続的コネクションの切断タイムアウト時間が、利用者の機器操作や機器の状態変化の間隔に比べて一般に短いので、操作や状態変化によって通信の必要が発生した時には、既に持続的コネクションが切断されており、持続的コネクションの効果が得られない場合がある。

【 0 0 2 2 】

持続的コネクションの効果を最大限に活かすためには、タイムアウトにより切断されないようにダミーの通信を行えばよいが、利用者の機器操作や機器の状態変化の間隔が長い場合、ダミーの通信量が増加するという問題があった。

【 0 0 2 3 】

そこで、本発明は、ファイアウォールの内部に操作対象機器（被操作装置）を設置しても、インターネットに接続された操作端末からファイアウォールを越えて操作できる遠隔操作システム、遠隔操作方法、被操作装置、操作端末、プログラムおよび記憶媒体を提供することを目的とする。

【 0 0 2 4 】

【課題を解決するための手段】

上記目的を達成するために、本発明の遠隔操作システムは、ネットワークを介して被操作装置および操作端末が接続され、前記操作端末で作成された操作情報を用いて前記被操作装置を操作する遠隔操作システムであって、前記被操作装置は、前記操作端末への通信を発動し、前記操作情報を前記操作端末に要求する操作情報要求手段を備え、前記操作端末は、前記発動された通信に応じ、所定時間だけ操作の発生を待機した後、前記要求された操作情報を前記被操作装置に返す

操作情報応答手段を備えたことを特徴とする。

【0025】

また、前記被操作装置は、前記操作情報の要求とは別に、前記操作端末への通信を発動し、該被操作装置の状態情報を前記操作端末に通知する状態情報通知手段を備え、前記操作端末は、前記発動された通信に応じ、待機することなく、前記通知された状態情報の応答を前記被操作装置に返す状態情報通知応答手段を備えたことを特徴とする。

【0026】

さらに、前記操作情報要求手段は、前記操作端末への通信を発動して前記被操作装置の状態情報を通知するとともに、該通信を用いてその応答としての前記操作情報を要求し、前記操作情報応答手段は、前記通知された状態情報の応答として、前記通信を用いて前記要求された操作情報を返すことを特徴とする。

【0027】

本発明の遠隔操作方法は、操作端末で作成された操作情報を用い、ネットワークを介して被操作装置を操作する遠隔操作方法であって、前記被操作装置から前記操作端末へ通信を発動し、前記操作情報を前記操作端末に要求する操作情報要求工程と、前記発動された通信に応じ、所定時間だけ操作の発生を待機した後、前記操作端末から前記要求された操作情報を前記被操作装置に返す操作情報応答工程とを有することを特徴とする。

【0028】

本発明の被操作装置は、ネットワークに接続された操作端末から操作される被操作装置であって、前記操作端末への通信を発動し、状態情報を通知する状態情報通知手段と、前記操作端末への通信を発動し、操作情報を要求する操作情報要求手段とを備え、前記操作端末は、前記被操作装置からの通信の発動に応じ、前記通知された状態情報の応答を待機することなく返す一方、前記要求された操作情報を所定時間待機した後に返すことを特徴とする。

【0029】

本発明の操作端末は、ネットワークに接続された被操作装置を操作する操作端末であって、前記被操作装置からの通信の発動に応じ、前記被操作装置から通知

された状態情報の応答を待機することなく前記被操作装置に返す状態情報通知応答手段と、前記被操作装置からの通信コネクションの発動に応じ、前記被操作装置から要求された操作情報を、所定時間待機した後に前記被操作装置に返す操作情報応答手段とを備えたことを特徴とする。

#### 【0030】

##### 【発明の実施の形態】

本発明の遠隔操作システム、遠隔操作方法、被操作装置、操作端末、プログラムおよび記憶媒体の実施の形態について図面を参照しながら説明する。

#### 【0031】

##### 〔第1の実施形態〕

図1は第1の実施形態における遠隔操作システムの構成を示す図である。図において、100はインターネットである。インターネット100には、ファイアウォール111を介してLAN110、および操作端末130が接続されている。また、LAN110には、被操作装置120が接続されている。ファイアウォール111には、プロキシサーバ112が導入されている。

#### 【0032】

被操作装置120は、CPU、メモリ（ROM、RAM）およびネットワークインタフェースを有するパーソナルコンピュータ、ワークステーションなどのコンピュータと同等の機能を有する装置であり、後述する状態通知プログラム141、操作データ収集プログラム142および制御プログラム143を有する。また、被操作装置120には、操作対象機器として、ON/OFFが操作される照明122、および開閉位置が操作されるブラインド123が接続されている。

#### 【0033】

コントローラ121は、制御プログラム143からの制御指令に応じて、照明122の点灯・消灯を制御したり、ブラインド123の上げ下げを制御するとともに、照明122の点灯状態やブラインド123の開閉位置などの状態情報を状態通知プログラム141に回答する。

#### 【0034】

操作端末130は、被操作装置120と同様、CPU、メモリ（ROM、RA

M) およびネットワークインタフェースを有するパーソナルコンピュータ、ワークステーションなどのコンピュータと同等の機能を有する装置であり、後述するHTTPサーバプログラム151、操作表示CGIプログラム152、操作CGIプログラム153、操作表示GUIプログラム154を有する。操作端末130には、表示装置131、およびマウスなどの操作入力装置132が接続されている。

#### 【0035】

操作端末130上で動作するHTTPサーバプログラム151は、HTTPリクエストに対する応答、CGI (Common Gateway Interface) プログラムの実行など、一般的なWWWサーバ機能を有するプログラムである。そして、操作端末130は、いわゆるWWWサーバのように振る舞う。

#### 【0036】

一方、被操作装置120では、状態通知プログラム141および操作データ収集プログラム142がHTTPサーバプログラム151にアクセスすることにより、操作端末130に向けて状態通知および操作データ収集依頼が行われる。操作表示GUIプログラム154は、被操作装置120から送られた装置の状態を表示装置131に表示し、利用者の操作対象機器に対する操作を操作入力装置132から受け取り、被操作装置120に送るためのGUIを実現する。

#### 【0037】

上記構成を有する遠隔操作システムの動作を示す。図2は遠隔操作システムの動作シーケンスを示す図である。図においては、状態通知サイクル、操作伝達サイクルの2本の通信コネクションを使用し、操作対象機器の状態通知および操作端末130からの操作情報の収集が行われる。

#### 【0038】

状態通知サイクルでは、被操作装置120上の状態通知プログラム141が操作端末130上の操作表示CGIプログラム152をアクセスすることで、状態通知が行われる。例えば、被操作装置120が状態情報S0、S1を操作端末130に通知すると、操作端末130側では、送られた状態情報を基に操作画面の表示が行われる。

## 【 0 0 3 9 】

一方、操作伝達サイクルでは、被操作装置 1 2 0 上の操作データ収集プログラム 1 4 2 が操作端末 1 3 0 上の操作 C G I プログラム 1 5 3 をアクセスすることで、被操作装置 1 2 0 から操作端末 1 3 0 に操作伝達要求が行われる。例えば、操作伝達 E 0、E 1 を収集すると、被操作装置 1 2 0 側では、収集した操作伝達情報を基に操作対象機器の操作が行われる。これらの状態通知サイクルおよび操作伝達サイクルは並列に動作する。

## 【 0 0 4 0 】

## 〔状態通知プログラム 1 4 1〕

つぎに、被操作装置 1 2 0 における状態通知サイクルを示す。図 3 は状態通知プログラム 1 4 1 の処理手順を示すフローチャートである。この処理は、被操作装置 1 2 0 内の C P U（図示せず）が同じく被操作装置 1 2 0 内の R O M（図示せず）に記憶された状態通知プログラム 1 4 1 を実行することによって行われる。

## 【 0 0 4 1 】

まず、操作対象機器に状態変化があるまで待つ（ステップ S 3 0 1）。この処理は、例えば、制御プログラム 1 4 3 が、操作対象機器に状態変化があったとき、共有メモリ上に確保された状態変化フラグ（図示せず）を O N にしておき、状態通知プログラム 1 4 1 が、このフラグを監視し、状態変化を検知した場合、状態変化フラグを O F F にする処理である。尚、共有メモリの代わりにファイルを共有してもよいし、通信ポートを利用し、それぞれのプログラム間でプロセス間通信を行うことで情報を共有してもよい。

## 【 0 0 4 2 】

ステップ S 3 0 1 で状態変化があった場合、照明 1 2 2 の点灯状態およびブラインド 1 2 3 の開閉位置の状態情報を収集する（ステップ S 3 0 2）。この処理は、コントローラ 1 2 1 の状態を入力する制御プログラム 1 4 3 に対して情報入力要求を行うことによって実現される。尚、制御プログラム 1 4 3 は、操作対象機器に状態変化があった際、その状態情報を格納するバッファを設け、状態通知プログラム 1 4 1 がこのバッファを監視することによって実現してもよい。



## 【0043】

HTTPのPOSTコマンドを用いて、状態情報に応じた送信情報を作成する（ステップS303）。例えば、照明122がOFFの状態であり、ブラインド123が70%の開閉位置であり、操作端末130における操作表示CGIプログラム152のURLがHTTP://foo.bar.co.jp/CGI-bin/newstatusである場合、以下のようなコマンドを作成する。

```
「POST HTTP://foo.bar.co.jp/CGI-bin/newstatus HTTP/1.0
```

```
Content-Length: 21
```

```
¥r¥n
```

```
light=OFF¥r¥n
```

```
blind=70¥r¥n」
```

そして、通信回線を確保し（ステップS304）、作成された情報を、操作端末130のHTTPサーバプログラム151に向けて送信する（ステップS305）。このように、ストリーム指向の接続で、ファイアウォールで守られた内部のLANからインターネット上への機器に接続要求を行う場合、ファイアウォールの通過を認めるように設定されることが多いので、通信が可能になる。HTTPサーバプログラム151は、この情報を受け取り、操作表示CGIプログラム152を呼び出す。操作表示CGIプログラム152は、受け取った状態情報を基に、操作対象機器の操作画面の更新を要求し、直ちに応答を返す。この詳細については後述する。

## 【0044】

状態通知プログラム141は、この応答が受信されるまで所定時間待つ（ステップS306、S308）。所定時間待っても応答がない場合、エラーフラグをONにセットする（ステップS307）。エラーフラグがONにセットされた場合、あるいは所定時間内に操作表示CGIプログラム152からの応答が受信された場合、回線をクローズする（ステップS309）。

## 【0045】

そして、エラーフラグがONにセットされたか否かを判別し（ステップS31

0)、エラーフラグがONにセットされている場合、このプログラムを終了し、そうでない場合、ステップS301に戻って再び状態変化を待つ。

【0046】

〔操作表示CGIプログラム152〕

つぎに、状態通知サイクルにおける操作端末130の処理を示す。予め操作端末130上で動作しているHTTPサーバプログラム151に対し、URLパスである「/CGI-bin/newstatus」が要求された場合、操作表示CGIプログラム152が実行されるように設定しておく。このため、上記要求を被操作装置120から受け取ると、操作表示CGIプログラム152が起動する。

【0047】

図4は操作表示CGIプログラム152の処理手順を示すフローチャートである。この処理は、操作端末130内のCPU（図示せず）が同じく操作端末130内のROM（図示せず）に格納された操作表示CGIプログラム152を実行することによって行われる。

【0048】

まず、被操作装置120から送られた状態情報を読み込み、その情報を共有メモリ領域に確保した情報交換テーブル401の中の状態欄に書き込む（ステップS401）。図5は情報交換テーブル401を示す図である。本実施形態では、照明がOFFであり、ブラインド開閉位置が70%であるので、状態欄のlight欄に「OFF」、状態欄のblind欄に「70」が書き込まれる。

【0049】

情報交換テーブル401は、操作表示GUIプログラム154から読み書き可能であり、この操作表示GUIプログラム154に対し、操作表示更新要求を行うことで操作画面が更新される。尚、共有メモリの代わりにファイルを共有してもよいし、通信ポートを利用し、それぞれのプログラム間でプロセス間通信を行い、情報の共有を行ってもよい。また、操作表示GUIプログラム154を利用する代わりに、操作表示CGIプログラム152に操作画面の表示・更新機能を持たせてもよい。

## 【0050】

そして、ACK応答情報を作成し、標準出力に出力する（ステップS402）。HTTPサーバプログラム151によって被操作装置120の状態通知プログラム141に応答が返される。例えば、以下のような応答情報を標準出力に出力する。

```
「Content-Type:  application/x-newstat  
us  
¥r¥n  
ok¥r¥n」
```

この結果、HTTPサーバプログラム151によって必要な他の情報が付加され、この例では、次のような応答が返される。

```
「HTTP/1.0  200  OK  
Content-Type:  application/x-newstatu  
s  
Content-Length:  4  
¥r¥n  
ok¥r¥n」
```

## 〔操作データ収集プログラム142〕

つぎに、被操作装置120における操作伝達サイクルの処理を示す。図6は操作データ収集プログラム142の処理手順を示すフローチャートである。この処理は、被操作装置120内のCPU（図示せず）が同じく被操作装置120内のROM（図示せず）に記憶された操作データ収集プログラム142を実行することによって行われる。

## 【0051】

まず、回線を確保する（ステップS501）。操作端末130のHTTPサーバプログラム151に向けて操作伝達要求コマンドを送信する（ステップS502）。例えば、操作端末130における操作CGIプログラム153のURLがHTTP://foo.bar.co.jp/CGI-bin/getoperationである場合、以下のようなコマンドを送信する。

```
「POST HTTP://foo.bar.co.jp/CGI-bin/getoperation HTTP/1.0
Content-Length: 6
¥r¥n
none¥r¥n」
```

HTTPサーバプログラム151は、この情報を受け取り、操作CGIプログラム153を呼び出す。操作CGIプログラム153は、操作イベントを待ち、イベントが発生した場合、操作情報を返送する。この詳細については後述する。

【0052】

操作データ収集プログラム142は、この応答が受信されるまで所定時間待つ（ステップS503、S505）。所定時間待っても応答がない場合、エラーフラグをONにセットし（ステップS504）、ステップS508の処理に移行する。

【0053】

一方、所定時間内に操作CGIプログラム153からの応答が受信された場合、受信データをチェックし、受信したデータが操作伝達であるか否かを判別する（ステップS506）。受信したデータが操作伝達である場合、操作情報を操作データキュー（図示せず）に追加する（ステップS507）。例えば、照明122をONにし、ブラインド123の開閉位置を80%にするように、操作端末側で指示された場合、以下のような応答が受信される。

```
「HTTP/1.0 200 OK
Content-Type: application/x-getoperation
Content-Length: 20
¥r¥n
light=ON¥r¥n
blind=80¥r¥n」
```

被操作装置120では、制御プログラム143は、イベントドリブンで動作し、操作データ収集プログラム142により、操作データキューの情報が書き換え

られた場合、操作要求イベントを発生する。操作要求イベントの発生が通知されると、本実施形態では、制御プログラム143は、照明122の点灯、およびブラインド123の開閉位置を80%にする操作情報を、操作データキューから取り出し、コントローラ121を制御し、照明122およびブラインド123を要求通りの状態に変更する。

#### 【0054】

そして、操作データ収集プログラム142は、回線をクローズする（ステップS508）。エラーフラグがONにセットされているか否かを判別し（ステップS509）、エラーフラグがONにセットされている場合、このプログラムを終了し、そうでない場合、ステップS501に戻り、同様の処理を繰り返す。

#### 【0055】

##### 〔操作CGIプログラム153〕

つぎに、操作伝達サイクルにおける操作端末130の処理を示す。状態通知サイクルと同様、予め操作端末130上で動作しているHTTPサーバプログラム151に対し、URLパスである「/CGI-bin/getoperation」が要求された場合、操作CGIプログラム153が実行されるように設定しておく。このため、このような要求を被操作装置120から受け取ると、操作CGIプログラム153が起動する。

#### 【0056】

図7は操作CGIプログラム153の処理手順を示すフローチャートである。この処理は、操作端末130内のCPU（図示せず）が同じく操作端末130内のROM（図示せず）に格納された操作CGIプログラム153を実行することによって行われる。

#### 【0057】

操作表示CGIプログラム152では、被操作装置120側に向けて直ちに応答を返したが、操作CGIプログラム153では、状態通知サイクルがHTTPリクエストのタイムアウトによる回線の切断が生じないような最大時間まで操作イベントを待つようにする。

#### 【0058】

まず、所定時間が経過するか（ステップS601）、あるいは操作が行われ、情報交換テーブル401の要求欄が更新されるまで待つ（ステップS602）。例えば、HTTPリクエストのタイムアウト時間から往復のネットワーク遅延時間を差し引いた時間より短い時間だけ操作イベントを待つようにする。

## 【0059】

操作が検出された場合、更新された要求を情報交換テーブル401の中の要求欄から読み取る（ステップS603）。例えば、照明122をONにし、ブラインド123を80%にするといった情報を読み込む。また、読み込んだ操作要求情報に応じた応答情報を作成し、標準出力に出力する（ステップS604）。HTTPサーバプログラム151によって被操作装置120上の操作データ収集プログラム142に応答を返す（ステップS605）。例えば、以下のような応答情報を標準出力に出力する。

```
「Content-Type:  application/x-getoper
a t i o n
¥ r ¥ n
l i g h t = O N ¥ r ¥ n
b l i n d = 8 0 ¥ r ¥ n」
```

HTTPサーバプログラム151によって必要な他の情報が付加され、操作データ収集プログラム142の応答情報で示したような情報が返される。

## 【0060】

一方、ステップS601で所定時間経過した場合、つまりユーザの操作がなかった場合、内容をnoneとして以下のような応答情報を標準出力に出力する。

```
「Content-Type:  application/x-getoper
a t i o n
¥ r ¥ n
n o n e ¥ r ¥ n」
```

このような構成を有することにより、第1の実施形態では、ファイアウォールの内部に被操作装置120を設置しても、インターネットに操作端末130を接続し、そこから対象機器を操作することが可能となる。

## 【 0 0 6 1 】

また、状態通知の応答を直ぐに返し、次の状態変化通知に備えると共に、操作伝達要求の場合、操作イベントを待つようにすることで、操作があった場合、直ぐに操作情報を通知することが可能となる。

## 【 0 0 6 2 】

## 〔第 2 の実施形態〕

前記第 1 の実施形態では、状態通知と操作伝達に専用の通信コネクションを設けていたが、本発明はこれに限るわけではない。第 2 の実施形態では、状態通知を行った同じ通信コネクション上で操作イベントを待つ場合を示す。

## 【 0 0 6 3 】

第 2 の実施形態の遠隔操作システムは、前記第 1 の実施形態のシステム構成（図 1 参照）において、状態通知プログラム 1 4 1 および操作データ収集プログラム 1 4 2 を、1 つの操作情報交換プログラム 7 4 1（図示せず）に置き換え、また操作表示 CGI プログラム 1 5 2 および操作 CGI プログラム 1 5 3 を、1 つの操作表示応答 CGI プログラム 7 5 2（図示せず）に置き換えることにより実現される。尚、前記第 1 の実施形態と同一の構成および処理については、同一の符号を付すことにより、その説明を省略する。

## 【 0 0 6 4 】

図 8 は第 2 の実施形態における動作シーケンスを示す図である。状態通知および操作伝達を共用する 2 本の通信コネクションを使用し、操作対象機器の状態通知および操作端末 1 3 0 からの操作情報の収集が行われる。

## 【 0 0 6 5 】

第 2 の実施形態では、1 本の通信コネクションでも状態通知および操作伝達を行うことができる点が前記第 1 の実施形態と異なる。すなわち、前記第 1 の実施形態と同様、被操作装置 1 2 0 で状態変化（例えば、S 0）を検出すると、状態変化を待っている片側の通信コネクションで操作端末 1 3 0 に向けて状態通知が行われる。そして、操作端末 1 3 0 側で操作が行われるのを待つ。操作イベントが検出されると、状態通知を行った通信コネクションで操作情報（例えば、E 0）が返送される。

## 【 0 0 6 6 】

被操作装置 1 2 0 側では、操作情報を受け取った後、どちらか一方の通信コネクションが状態変化待ちの状態である場合、今度は状態変化がなくても、「状態変化なし」という状態通知を行う。そして、収集した操作情報を基に、操作対象機器の操作が行われる。

## 【 0 0 6 7 】

状態通知を行った通信コネクションが操作を待っている間、被操作装置 1 2 0 で新たな状態変化（例えば、S 1）を検出すると、新たに通信コネクションが生成され、状態通知が行われる。操作端末 1 3 0 側では、新たな状態通知が到着すると、先に操作待ちを行っている通信コネクション側で「操作なし」という操作情報を返送し、後の通信コネクション側で引き続き操作を待つ。「操作なし」を受け取った通信コネクション側では、もう一方が操作伝達あるいは操作なしを受信するまで状態変化を待つ。

## 【 0 0 6 8 】

## 〔操作情報交換プログラム 7 4 1〕

つぎに、被操作装置 1 2 0 側の処理を示す。図 9 および図 1 0 は操作情報交換プログラム 7 4 1 の処理手順を示すフローチャートである。この処理は、被操作装置 1 2 0 内の CPU（図示せず）が同じく被操作装置 1 2 0 内の ROM（図示せず）に記憶された操作情報交換プログラム 7 4 1 を実行することによって行われる。

## 【 0 0 6 9 】

操作情報交換プログラム 7 4 1 は、3 つのカウンタ（図示せず）を有する。図 1 1 はカウンタテーブルを示す図である。第 1 のカウンタは、操作情報交換プログラム 7 4 1 が HTTP サーバプログラム 1 5 1 からの応答を待っている HTTP リクエスト要求数を数えるものであり、その初期値は値 0 である。このカウンタを要求カウンタと呼ぶ。

## 【 0 0 7 0 】

第 2 のカウンタは、送信する HTTP リクエストの順序を数えるものであり、リクエストデータ中にその値が埋め込まれる。その初期値は値 0 である。このカ



ウンタを送信カウンタと呼ぶ。

【0071】

第3のカウンタは、受信したHTTPレスポンスの処理順序を数えるものであり、レスポンスデータの中にその値が埋め込まれている。その初期値は値0である。このカウンタを受信カウンタと呼ぶ。

【0072】

本実施形態では、操作情報交換プログラム741は、同時に2つのプロセスで実行され、並列動作する。まず、共有メモリ領域に確保したカウンタテーブル801から要求カウンタを読み込む（ステップS801）。カウンタテーブル801には、要求カウンタ以外に、後述する送信カウンタおよび受信カウンタの値がセットされている。カウンタテーブル801の要求欄には、現在、状態通知を行って操作伝達待ちである通信コネクションの数が設定されている。つまり、要求カウンタが値0である場合、2つの操作情報交換プログラム741は共に状態変化待ちの状態である。値1の場合、片方が状態通知を行った後、操作待ち状態であることを示す。尚、要求、送信、受信のそれぞれのカウンタを別々に管理してもよく、また共有メモリの代わりにファイルを共有してもよい。

【0073】

要求カウンタのチェックを行い（ステップS802）、要求カウンタが値1である場合、状態変化が発生するまで待つ（ステップS803）。一方、要求カウンタが値0である場合、あるいはステップS803で状態変化があった場合、前記第1の実施形態と同様、照明122の点灯状態およびブラインド123の開閉位置の状態情報を収集し（ステップS804）、カウンタテーブル801から送信カウンタを読み込む（ステップS805）。

【0074】

そして、送信情報を作成する（ステップS806）。カウンタテーブル801の送信欄には、状態通知情報を送信した回数がセットされている。例えば、送信カウンタの値が「1」である場合、以下のようなコマンドを作成する。

```
「POST HTTP://foo.bar.co.jp/CGI-bin/newstatus HTTP/1.0
```

Content-Length: 37

¥r¥n

send\_counter=1¥r¥n

light=OFF¥r¥n

blind=70¥r¥n」

そして、送信カウンタの値を1つ増やす（ステップS807）。本実施形態では、カウンタテーブル801の送信欄に値2を書き込む。通信回線を確認し（ステップS808）、要求カウンタの値を1つ増やし（ステップS809）、カウンタテーブルの要求欄に書き込む。

#### 【0075】

尚、本実施形態では、2つの操作情報交換プログラム741が並列動作しているので、ステップS804～ステップS807の処理を行う場合、情報交換テーブル401、カウンタテーブル801をロックするなどの排他制御を行うことで、2重の情報収集、カウンタの抜け、だぶりなどが発生しないようにしておく。また、後述する応答受信の場合も、同様の排他制御を行う。

#### 【0076】

ステップS809の処理後、操作端末130のHTTPサーバプログラム151に向けてこの情報を送信する（ステップS810）。尚、要求カウンタの初期値を値0にしておくことで、操作情報交換プログラム741が最初に起動されたとき、状態変化を待たずに、操作対象機器の初期状態を操作端末103に通知しておくことができる。

#### 【0077】

HTTPサーバプログラム151は、この情報を受け取り、操作表示応答CGIプログラム752を呼び出す。操作表示応答CGIプログラム752は、受け取った情報を基に対象機器の操作画面を更新する。この詳細については後述する。

#### 【0078】

操作情報交換プログラム741は、操作表示応答CGIプログラム752からの応答を受信するまで所定時間待つ（ステップS811）。所定時間待っても応

答がない場合、エラーフラグをONにセットし（ステップS812）、ステップS819の処理に移行する。

#### 【0079】

一方、操作表示応答CGIプログラム752からの応答を受信した場合、カウンタテーブル801から受信カウンタを読み込み（ステップS814）、受信カウンタのチェックを行う（ステップS815）。カウンタテーブル801の受信欄には、既に受信したデータのカウンタ値、つまり次に処理すべきデータ番号より値1少ない値がセットされている。ここで、受け取ったデータ内に記述されているカウンタ番号が受信カウンタの値より値2以上大きい場合、データに抜けがあることが分かるので、もう一方の通信コネクションでデータ受信を行い、受信カウンタが更新されるまで待つ。

#### 【0080】

一方、ステップS815でカウンタ番号の値が正しい場合、受信データが操作要求情報であるか、あるいは操作なし情報であるかを判別する（ステップS816）。操作要求情報である場合、操作情報を操作データキュー（図示せず）に追加する（ステップS817）。例えば、以下のような応答が受信される。

```
「HTTP/1.0 200 OK
Content-Type: application/x-getoperation
Content-Length: 39
¥r¥n
receive_counter=1¥r¥n
light=ON¥r¥n
blind=80¥r¥n」
```

そして、受信カウンタの値を1つ増やす（ステップS818）。本実施形態では、カウンタテーブル801の受信欄に値1を書き込む。尚、ステップS816～ステップS818の処理を行う際、前述と同様、排他制御を行うようにする。また、制御プログラム143の動作については、前記第1の実施形態と同様である。

## 【0081】

この後、操作情報交換プログラム741は、回線をクローズし（ステップS819）、再度、要求カウンタを読み出し（ステップS820）、要求カウンタの値を1つ減らし、カウンタテーブル801の要求欄を更新する（ステップS821）。そして、エラーフラグがONにセットされているか否かを判別し（ステップS822）、エラーフラグがONにセットされている場合、この処理を終了し、そうでない場合、ステップS801に戻って同様の処理を繰り返す。

## 【0082】

## 〔操作表示応答CGIプログラム752〕

つぎに、操作端末130側の処理を示す。前記第1の実施形態と同様、予め操作端末130で動作しているHTTPサーバプログラム151に対し、URLパスである「/CGI-bin/newstatus」が要求された場合、操作表示応答CGIプログラム752が実行されるように設定しておく。このため、上記要求を被操作装置120から受け取ると、操作表示応答CGIプログラム752が起動する。

## 【0083】

本実施形態では、操作情報交換プログラム741が並列動作するので、同時に2つの要求を受け取る場合もある。このとき、操作表示応答プログラム752はそれぞれの要求毎に起動して並列動作する。

## 【0084】

図12および図13は操作表示応答CGIプログラム752の処理手順を示すフローチャートである。操作表示応答CGIプログラム752は、3つのカウンタ（図示せず）を有する。

## 【0085】

第1のカウンタは、操作情報交換プログラム741からのHTTPリクエストにより起動された操作表示応答CGIプログラム752の実行プロセスを数えるものであり、その初期値は値0である。このカウンタを要求カウンタと呼ぶ。

## 【0086】

第2のカウンタは、受信した状態情報の順序を数えるものであり、操作情報交

換プログラム 7 4 1 の送信カウンタと対を成す。その初期値は値 0 であり、リクエストデータ中にその値が埋め込まれる。このカウンタを受信カウンタと呼ぶ。

## 【 0 0 8 7 】

第 3 のカウンタは、送信する HTTP レスポンスの処理順序を数えるものであり、操作情報交換プログラム 7 4 1 の受信カウンタと対を成す。その初期値は値 0 であり、レスポンスデータ中にその値が埋め込まれる。このカウンタを受信カウンタと呼ぶ。

## 【 0 0 8 8 】

まず、共有メモリ領域に確保した前述したカウンタテーブル 8 0 1 と同等のカウンタテーブル（図示せず）から要求カウンタを読み込む（ステップ S 9 0 1）。カウンタテーブルには、前述と同様、要求カウンタ以外に、後述する送信カウンタおよび受信カウンタの値がセットされている。カウンタテーブルの要求欄には、現在、実行中の操作表示応答 CGI プログラム 7 5 2 の数が設定されている。そして、要求カウンタの値を 1 つ増やす（ステップ S 9 0 2）。

## 【 0 0 8 9 】

受信カウンタを読み込み（ステップ S 9 0 3）、被操作装置 1 2 0 から送られた状態情報の中のカウンタ番号と受信カウンタの番号とから処理すべき状態情報であるか否かを判別する（ステップ S 9 0 4）。カウンタ番号が正しい場合、受信カウンタの値を 1 つ増やす（ステップ S 9 0 5）。

## 【 0 0 9 0 】

そして、前記第 1 の実施形態と同様、操作表示 GUI プログラム 1 5 4 に操作表示更新要求を行う（ステップ S 9 0 6）。尚、ステップ S 9 0 5 およびステップ S 9 0 6 の処理を行う際、前述したような排他制御を行い、これらの処理が終わるまで並列動作している操作情報交換プログラムによる同じ処理ステップが行われないようにする。また、操作表示 GUI プログラムの処理については、前記第 1 の実施形態と同様である。

## 【 0 0 9 1 】

この後、所定時間が経過したか否かを判別し（ステップ S 9 0 7）、所定時間経過している場合、ステップ S 9 1 0 の処理に移行する。一方、所定時間が経過

していない場合、要求カウンタをチェックし（ステップS908）、要求カウンタが値2である場合、ステップS910の処理に移行する。一方、要求カウンタが値1である場合、操作イベントが検出されたか否かを判別する（ステップS909）。操作イベントが検出されない場合、ステップS907の処理に戻り、操作イベントが検出された場合、ステップS910の処理に移行する。

## 【0092】

このように、所定時間が経過するまで要求カウンタをチェックしながら、操作が行われ、前記第1の実施形態と同様に情報交換テーブル（図示せず）の要求欄が更新されるのを待つ。

## 【0093】

ステップS909で操作イベントが検出されると、更新された要求を、情報交換テーブルの中の要求欄から読み取る（ステップS910）。尚、操作イベントがない場合、「操作なし」として取り扱う。

## 【0094】

送信カウンタを読み込み（ステップS911）、送信カウンタの値を1つ増やす（ステップS912）。尚、ステップS910～ステップS912の処理を行う際、前述した排他制御を行うようにする。

## 【0095】

読み込んだ操作情報に応じて応答情報を作成し、標準出力に出力する（ステップS913）。HTTPサーバプログラム151によって被操作装置120上の操作情報交換プログラム741に向けて応答を返す（ステップS914）。例えば、以下のような応答情報を標準出力に出力する。

```
「Content-Type: application/x-getoper
ation
¥r¥n
receive_counter=1¥r¥n
light=ON¥r¥n
blind=80¥r¥n」
```

HTTPサーバプログラム151によって必要な他の情報が付加され、操作情

報交換プログラム741の応答情報で示したような情報が返される。

【0096】

一方、ステップS907で所定時間が経過した場合、つまりユーザの操作がなかった場合、あるいは、ステップS908で複数の操作待ち状態が検出された場合、内容をnoneとして以下のような応答情報（操作なし）を標準出力に出力する。

```
「Content-Type: application/x-getoper
ation
¥r¥n
receive_counter=1¥r¥n
none¥r¥n」
```

この後、要求カウンタの値を1つ減らす（ステップS915）。

【0097】

尚、本実施形態では、通信コネクションが2本である場合を示したが、3本以上の通信コネクションを用いてもよい。その場合、図9のステップS802の要求カウンタチェックルーチンにおいて、要求数が値0である場合、状態変化が無くても状態通知を行い、要求数が値1である場合、状態変化を待ち、要求数が値2である場合、どちらかの通信コネクションが切断されるのを待つという処理を行うようにすればよい。例えば、利用する通信コネクションがn本であり、操作イベントをm個（ $m < n$ ）のプロセスで待つ場合、要求数が値0から値 $m-1$ （ $m$ マイナス1）の場合、状態変化が無くても状態通知を行い、要求数が $m$ から $n-1$ （ $n$ マイナス1）の場合、状態変化を待ち、要求数が $n$ の場合、いずれかの通信コネクションが切断されるのを待つようにする。

【0098】

同様に、図12および図13のステップS908の要求カウンタチェックルーチンにおいて、要求数が値1である場合、操作イベントを待ち、要求数が値2である場合、操作待ちループを抜けるという処理を、要求数が値1から値 $m-1$ （ $m$ マイナス1）である場合、操作イベントを待ち、要求数が値 $m$ である場合、操作待ちループを抜けるようにする。

## 【0099】

このような構成を有することにより、第2の実施形態では、ファイアウォールの内部に被操作装置120を設置しても、インターネットに操作端末130を接続し、そこから対象機器を操作することが可能となる。

## 【0100】

また、1本の通信コネクションだけでも、状態通知および操作情報伝達を行うことができ、2本以上の通信コネクションを利用する場合であっても、2本目以降の通信コネクションは必要なときにしか使われないので、ネットワークの負荷を軽減することができる。

## 【0101】

## [第3の実施形態]

前記第1の実施形態では、状態通知サイクルおよび操作伝達サイクルに、それぞれ1本の通信コネクションを使用していたが、本発明はこれに限るわけではない。第3の実施形態では、状態通知サイクル、操作伝達サイクルのいずれか、または両方に複数の通信コネクションを使用する場合を示す。これにより、連続した状態変化または操作要求に迅速に対応することが可能である。

## 【0102】

第3の実施形態における遠隔操作システムは、前記第1の実施形態のシステム構成において、状態通知プログラム141を状態通知プログラム10141（図示せず）に、操作データ収集プログラム142を操作データ収集プログラム10142（図示せず）に、操作表示CGIプログラム152を操作表示CGIプログラム10152（図示せず）に、操作CGIプログラム153を操作CGIプログラム10153（図示せず）にそれぞれ変更することによって実現される。また、前記第1の実施形態と同一の構成および処理については、同一の符号を用いることにより、その説明を省略する。

## 【0103】

## [状態通知プログラム10141]

被操作装置120における状態通知サイクルの処理を示す。図14は第3の実施形態における状態通知プログラム10141の処理手順を示すフローチャート



である。この処理は、被操作装置 1 2 0 内の CPU（図示せず）が同じく被操作装置 1 2 0 内の ROM（図示せず）に記憶された状態通知プログラム 1 0 1 4 1 を実行することによって行われる。

#### 【0 1 0 4】

第 3 の実施形態では、状態通知プログラム 1 0 1 4 1 は、同時に 2 つ以上のプロセスで実行され、並列動作する。状態通知プログラム 1 0 1 4 1 は、前記第 2 の実施形態におけるステップ S 8 0 5、ステップ S 8 0 7 とそれぞれ同様の送信カウンタ読込処理（ステップ S 3 0 2 A）、送信カウンタ加算処理（ステップ S 3 0 3 A）が加わったことを除き、前記第 1 の実施形態における状態通知プログラム 1 4 1 と同じである。また、ステップ S 3 0 2 ～ステップ S 3 0 3 A の処理を行う際、前記第 2 の実施形態で示した排他制御を行うようにする。

#### 【0 1 0 5】

この結果、複数の状態通知プログラム 1 0 1 4 1 が並列動作する場合においても、送信データにデータ番号が付与されるので、操作端末 1 3 0 に到着した際、順番が逆になっても、後述する操作表示 CGI プログラム 1 0 1 5 2 によって正しい順に処理を行うことが可能となる。

#### 【0 1 0 6】

##### 〔操作表示 CGI プログラム 1 0 1 5 2〕

つぎに、状態通知サイクルにおける操作端末 1 3 0 の処理を示す。前記第 1 の実施形態と同様、予め操作端末 1 3 0 上で動作している HTTP サーバプログラム 1 5 1 に対し、URL パスである「/CGI-bin/newstatus」が要求された場合、操作表示 CGI プログラム 1 0 1 5 2 が実行されるように設定しておく。このため、上記要求を被操作装置 1 2 0 から受け取ると、操作表示 CGI プログラム 1 0 1 5 2 が起動する。

#### 【0 1 0 7】

図 1 5 は操作表示 CGI プログラム 1 0 1 5 2 の処理手順を示すフローチャートである。この処理は、操作端末 1 3 0 内の CPU（図示せず）が同じく操作端末 1 3 0 内の ROM（図示せず）に格納された操作表示 CGI プログラム 1 0 1 5 2 を実行することによって行われる。

## 【0108】

本実施形態では、状態通知プログラム10141が並列動作するので、同時に2つ以上の要求を受け取る場合もある。この場合、操作表示CGIプログラム10152がそれぞれの要求毎に起動し、並列動作する。

## 【0109】

操作表示CGIプログラム10152は、前記第2の実施形態におけるステップS903と同様の受信カウンタ読み込み処理（ステップS400A）、ステップS904と同様の受信カウンタチェック処理（ステップS400B）、ステップS905と同様の受信カウンタ加算処理（ステップS400C）が加わったことを除き、前記第1の実施形態における操作表示CGIプログラム152と同じである。

## 【0110】

尚、ステップS400CおよびステップS401の処理を行う際、前述した排他制御を行い、これらの処理が終わるまで、並列動作している操作表示CGIプログラム10152による同じ処理ステップが行われないようにする。

## 【0111】

この結果、複数の操作表示CGIプログラム10152が並列動作する場合でも、受信カウンタを管理し、受信データに付与されたデータ番号を確認することにより、正しい順に処理を行うことが可能となる。

## 【0112】

## 〔操作データ収集プログラム10142〕

つぎに、操作伝達サイクルにおける被操作装置120の処理を示す。図16は操作データ収集プログラム10142の処理手順を示すフローチャートである。この処理は、被操作装置120内のCPU（図示せず）が同じく被操作装置120内のROM（図示せず）に記憶された操作データ収集プログラム10142を実行することによって行われる。

## 【0113】

第3の実施形態では、操作データ収集プログラム10142は、同時に2つ以上のプロセスで実行され、並列動作する。

## 【0114】

操作データ収集プログラム10142は、前記第2の実施形態におけるステップS814と同様の受信カウンタ読込み処理（ステップS505A）、ステップS815と同様の受信カウンタチェック処理（ステップS505B）、ステップS818と同様の受信カウンタ加算処理（ステップS507A）が加わったことを除き、前記第1の実施形態における操作データ収集プログラム142と同じである。尚、ステップS506～ステップS507Aの処理を行う際、前記第2の実施形態で示した排他制御を行うようにする。

## 【0115】

この結果、複数の操作データ収集プログラム10142が並列動作する場合でも、受信カウンタを管理し、受信データに付与されたデータ番号を確認することにより、正しい順に処理を行うことが可能となる。

## 【0116】

## 〔操作CGIプログラム10153〕

つぎに、操作端末130における操作伝達サイクルの処理を示す。前記第1の実施形態と同様、予め操作端末130上で動作しているHTTPサーバプログラム151に対し、URLパスである「/CGI-bin/getoperation」が要求された場合、操作CGIプログラム10153が実行されるように設定しておく。このため、上記要求を被操作装置120から受け取ると、操作CGIプログラム10153が起動する。

## 【0117】

図17は操作CGIプログラム10153の処理手順を示すフローチャートである。この処理は、操作端末130内のCPU（図示せず）が同じく操作端末130内のROM（図示せず）に格納された操作CGIプログラム10153を実行することによって行われる。

## 【0118】

本実施形態では、操作データ収集プログラム10142が並列動作するので、同時に2つ以上の要求を受け取る場合もある。この場合、操作CGIプログラム10153がそれぞれの要求毎に起動し、並列動作する。

## 【0119】

操作CGIプログラム10153は、第2の実施形態におけるステップS901と同様の要求カウンタ読み込み処理（ステップS600A）、ステップS902と同様の要求カウンタ加算処理（ステップS600B）、ステップS908と同様の要求カウンタチェック処理（ステップS601A）、ステップS911と同様の送信カウンタ読み込み処理（ステップS603A）、ステップS912と同様の送信カウンタ加算処理（ステップS603B）、ステップS915と同様の要求カウンタ減算処理（ステップS605A）が加わったことを除き、前記第1の実施形態における操作CGIプログラム153と同じである。尚、この場合、ステップS603～ステップS603Bの処理を行う際、第2の実施形態で示した排他制御を行うようにする。

## 【0120】

この結果、複数の操作CGIプログラム10153が並列に動作する場合でも、送信データにデータ番号が付与されるので、被操作装置120に到着した際に順番が逆になっていても、前述の操作データ収集プログラム10142で正しい順に処理を行うことが可能となる。

## 【0121】

このような構成を有することにより、第3の実施形態では、状態通知サイクル、操作伝達サイクルのいずれか、または両方を複数の通信コネクションでも利用することができ、これにより、状態変化や操作要求が短い時間内に連続して発生するような場合でも、状態通知や操作伝達が遅れないようにすることができる。

## 【0122】

## 〔第4の実施形態〕

第4の実施形態は、前記第2の実施形態と異なり、操作端末との通信を司る専用のプログラムが被操作装置に置かれ、持続的通信コネクションを用いて処理を行うことを特徴する。

## 【0123】

図18は第4の実施形態における遠隔操作システムの構成を示す図である。図において、100～132、143、151、153の各部は前記第1の実施形

態と同一であるので、その説明を省略する。ここで、本実施形態のHTTPサーバプログラム151は、持続的コネクションおよびパイプライン処理に対応し、特にパイプライン処理では、HTTPの規定にしたがってリクエストの到着順にレスポンスを返すことを保証する機能を有する。

#### 【0124】

また、操作情報交換プログラム201は、本実施形態に特有のものであり、前記第2の実施形態における操作情報交換プログラム741の代りに、被操作装置120に置かれて動作する。通信制御プログラム202は、新たに被操作装置120に置かれて動作する。

#### 【0125】

一方、操作端末130側では、前記第2の実施形態における操作表示応答CGIプログラム752の代りに、本実施形態に特有の操作表示応答CGIプログラム211が置かれて動作する。以下、操作情報交換プログラム201、通信制御プログラム202および操作表示応答CGIプログラム211を示す。

#### 【0126】

##### [通信制御プログラム202]

通信制御プログラム202は、操作情報交換プログラム201を子プロセスとして起動・管理するとともに、操作端末130上のHTTPサーバプログラム151との間で持続的コネクションを確立・維持し、持続的コネクションにおける状態情報および操作情報の通信制御を司る。また、通信制御プログラム202は、2種類のタイマ、そのタイムアウト値、1つのカウンタ、および2つの状態フラグを有する。

#### 【0127】

第1のタイマは、持続的コネクションのタイムアウトを防ぐために用いられ、HTTPレスポンスの受信から次のHTTPリクエストの発信までの時間を計測する。そのタイムアウト値は、一般的なプロキシサーバプログラムの持続的コネクションのタイムアウト時間よりわずかに短く設定される。この第1のタイマは、通信制御プログラム202に対して1つ割り当てられる。以下、このタイマを持続的コネクションタイムアウトタイマと呼ぶ。

## 【0128】

第2のタイマは、HTTPサーバプログラム151、プロキシサーバプログラム112、ネットワークのいずれかに障害が発生した際、通信制御プログラム202が無限の待ち状態に陥らないようにするためのものであり、HTTPリクエストとして状態情報を送信してからそのリクエストに対応するHTTPレスポンスが返るまでの時間を計測する。そのタイムアウト値は、一般のHTTPにおけるタイムアウト値と同等に設定されており、固定である。HTTPリクエスト1つに対し、1つのタイマが割り当てられ、リクエストの送信とともに計測がスタートし、レスポンスの受信とともに計測を終了し、割り当てが解放される。以下、このタイマをHTTPタイムアウトタイマと呼ぶ。

## 【0129】

カウンタは、通信制御プログラム202がHTTPサーバプログラム151からの応答を待っているHTTPリクエスト数を数えるものであり、その初期値は値0である。以下、このカウンタを応答待ちカウンタと呼ぶ。

## 【0130】

2つの状態フラグは、プログラム間の共有メモリ領域に確保されており、操作情報交換プログラム201群との間でデータの送信要求および受信要求を伝えるために使用される。初期値はともに値0である。以下、それぞれ送信待ちフラグ、受信待ちフラグと呼ぶ。

## 【0131】

図19および図20は通信制御プログラム202の処理手順を示すフローチャートである。この処理は、被操作装置120内のCPU（図示せず）が同じく被操作装置120内のROM（図示せず）に記憶された通信制御プログラム202を実行することによって行われる。

## 【0132】

通信制御プログラム202は、起動すると、自身の子プロセスとして操作情報交換プログラム201を事前に設定された所定数起動する（ステップS1500）。ここでは、通信制御プログラム202がn個起動する場合を示す（nは2以上の整数）。そして、操作情報交換プログラム201群と自身との間にプロセス

間の通信路を確立する（ステップ S 1 5 1 0）。

【 0 1 3 3 】

つぎに、HTTPサーバプログラム 1 5 1 との間に持続的コネクション（persistent connection）を確立する（ステップ S 1 5 2 0）。ここでは、利用するHTTPのバージョンに応じた方法により、持続的コネクションを確立する。

【 0 1 3 4 】

尚、この持続的コネクションは、ファイアウォール 1 1 1 上のプロキシサーバプログラム 1 1 2 との間で確立され、プロキシサーバプログラム 1 1 2 と HTTPサーバプログラム 1 5 1 との間では、これとは独立した持続的コネクションがプロキシサーバプログラム 1 1 2 によって確立される。

【 0 1 3 5 】

この後、持続的コネクションタイムアウトタイマをクリアする（ステップ S 1 5 3 0）。

【 0 1 3 6 】

上記処理により、通信制御プログラム 2 0 2 は初期設定を終え、これ以降、操作情報交換プログラム 2 0 1 群と HTTPサーバ 1 5 1 との間でデータ交換を行う。まず、操作情報交換プログラム 2 0 1 群から HTTPサーバ 1 5 1 へのデータ送信要求が起こされているか否かを調べる（ステップ S 1 5 4 0）。このチェックは、操作情報交換プログラム 2 0 1 の説明において詳述するが、簡単に説明すると、送信待ちフラグの値を調べ、この値が「0」でない場合、そのフラグの値が示すプロセス ID を持つ子プロセスである操作情報交換プログラム 2 0 1 が送信を要求していると判別する。

【 0 1 3 7 】

送信要求がなかった場合、持続的コネクションタイムアウトタイマが所定タイムアウト時間に達しているか否かを判別する（ステップ S 1 6 3 0）。所定タイムアウト時間に達していた場合、持続的コネクションを維持するためにのみ、ダミーの HTTP リクエストを作成し、これを HTTPサーバ 1 5 1 に送信し（ステップ S 1 6 4 0）、そのレスポンスを受け取る（ステップ S 1 6 5 0）。そし

て、ステップS1540の処理に戻る。

【0138】

このリクエストは、操作端末130上のHTTPサーバプログラム151が応答を返すものであれば何でもよく、例えば、HEAD HTTP://foo.bar.co.jp/ HTTP/1.0のような転送データ量が少ないものであることが望ましい。受け取ったデータは不要であるので、破棄する。そして、レスポンスを受け取ると、ステップS1530の処理に戻る。一方、ステップS1630で所定タイムアウト時間に達していなかった場合、ステップS1540の処理に戻る。

【0139】

一方、ステップS1540で要求があった場合、該当する子プロセスからプロセス間通信路を用いて送信データを受け取り、これをHTTPサーバプログラム151にHTTPリクエストとして持続的コネクション上で転送する（ステップS1550）。応答待ちカウンタをインクリメントする（ステップS1560）。

【0140】

パイプライン処理機能を活かし、レスポンスが返される前にデータ送信要求が起こされているか否かを再度調べる（ステップS1570）。データ送信要求があった場合、ステップS1550の処理に戻る。

【0141】

一方、送信要求がなかった場合、HTTPタイムアウトタイマのいずれかがタイムアウトしていないか否かを調べる（ステップS1580）。タイムアウトしていた場合、エラー処理を行い（ステップS1660）、子プロセスである操作情報交換プログラム201群を割り込み処理等で終了させ（ステップS1670）、HTTPサーバプログラム151との間の持続的コネクションをクローズし（ステップS1680）、この通信制御プログラム202の処理を終了する。

【0142】

一方、ステップS1580でタイムアウトしていない場合、応答データがHTTPサーバ151から送られてきているか否かを調べる（ステップS1590）



。応答データを受信していた場合、データの読み出しを待っているプロセスを調べ、そのプロセスにデータを転送する（ステップS1600）。この処理は、操作情報交換プログラム201の説明の際に詳述するが、簡単に説明すると、受信待ちフラグの値を調べ、このフラグの値が示すプロセスIDを持つ子プロセスである操作情報交換プログラム201にデータを転送する。

## 【0143】

応答待ちカウンタをデクリメントし（ステップS1610）、カウンタの値が「0」になったか否かを判別する（ステップS1620）。値0であった場合、HTTPリクエストが残っていないので、ステップS1530の処理に戻り、パイプライン処理を一旦、終了する。一方、ステップS1620で値0でない場合、引き続きパイプライン処理を行うために、ステップS1570の処理に戻る。また、ステップS1590で応答データが到着していなかった場合、ステップS1570の処理に戻る。

## 【0144】

## 〔操作情報交換プログラム201〕

操作情報交換プログラム201の処理は、通信制御プログラム202との関連部分を除き、前記第2の実施形態における操作情報交換プログラム741と基本的に同一であるので、操作情報交換プログラム741と同一のステップ処理については、その説明を省略する。

## 【0145】

操作情報交換プログラム201は、プログラム間の共有メモリ領域に2つのカウンタ、2つのキューおよび2つの状態フラグを有する。

## 【0146】

第1のカウンタは、同時に動作する操作情報交換プログラム201間で共有されるものであり、状態通知・操作伝達要求を行い、現在応答を待っている数をカウントする。初期値は「0」である。以下、このカウンタを要求カウンタと呼ぶ。

## 【0147】

第2のカウンタも同時に動作する操作情報交換プログラム201間で共有され

るものであり、これまでに状態通知を送信した回数をカウントする。初期値は、操作表示応答CGIプログラム211の受信カウンタの初期値と同じである。以下、このカウンタを送信カウンタと呼ぶ。

## 【0148】

第1のキューも同時に動作する操作情報交換プログラム201間で共有されるものであり、状態変化待ちを行っているプログラムのプロセスIDを格納する。初期値は空である。以下、このキューを状態変化待ちキューと呼ぶ。

## 【0149】

第2のキューは、制御プログラム143との間で共有されるものであり、操作情報データの受け渡しに用いられる。以下、このキューを操作データキューと呼ぶ。

## 【0150】

2つの状態フラグは、通信制御プログラム202で示した送信待ちフラグおよび受信待ちフラグである。

## 【0151】

図21および図22は操作情報交換プログラム201の処理手順を示すフローチャートである。この処理は、被操作装置120内のCPU（図示せず）が同じく被操作装置120内のROM（図示せず）に記憶された操作情報交換プログラム201を実行することによって行われる。

## 【0152】

操作情報交換プログラム201は、通信制御プログラム202によって所定数起動される。ここでは、 $n$ 個起動される場合を示す。起動すると、まず、要求カウンタの値を読み込む（ステップS2000）。要求カウンタの値 $i$ を判別する（ステップS2010）。

## 【0153】

ここで、前記第2の実施形態で示したように、操作イベントを $m$ 個のプログラムで待つ場合（ただし、 $m$ は1以上 $n$ 未満の整数）、要求カウンタの値 $i$ が「0」から「 $m-1$ 」である場合、状態変化がなくとも状態通知を行うために、ステップS2050の処理に移行する。

## 【0154】

一方、要求カウンタの値  $i$  が「 $n$ 」である場合、いずれかの要求が戻るまで待つために、ステップ S 2 0 0 0 の処理に戻る。また一方、要求カウンタの値  $i$  が「 $m$ 」から「 $n-1$ 」である場合、状態変化を待つために、自身のプロセス ID を状態変化待ちキューに入れる（ステップ S 2 0 2 0）。

## 【0155】

そして、自身がキューの先頭に出るまで待ち（ステップ S 2 0 3 0）、先頭になった場合、状態変化が起るまで待ち（ステップ S 2 0 4 0）、状態変化が起った場合、状態変化待ちキューから自身の ID を抜く（ステップ S 2 0 4 5）。

## 【0156】

排他制御を行うために、送信待ちフラグが「0」になるのを待つ（ステップ S 2 0 5 0）。操作情報交換プログラム 2 0 1 は、自身のプロセス ID を送信待ちフラグに入れて「0」以外の値とし、排他制御を実現する（ステップ S 2 0 6 0）。

## 【0157】

排他制御に入った場合、前述したステップ S 8 0 4 の処理と同様、状態情報を収集し（ステップ S 2 0 7 0）、前述したステップ S 8 0 5 の処理と同様、送信カウンタの値を読み込み（ステップ S 2 0 8 0）、前述したステップ S 8 0 6 の処理と同様、送信情報を作成する（ステップ S 2 0 9 0）。

## 【0158】

この後、要求カウンタおよび送信カウンタの値をそれぞれインクリメントし（ステップ S 2 1 0 0）、通信制御プログラム 2 0 2 に作成した送信情報を渡して HTTP サーバプログラム 1 5 1 への送信を依頼した後（ステップ S 2 1 1 0）、送信待ちフラグを値 0 に戻して排他制御を抜ける（ステップ S 2 1 2 0）。

## 【0159】

そして、HTTP サーバプログラム 1 5 1 からの応答データの受信に際しても排他制御を行う。同様に、受信待ちフラグの値が「0」になるのを待つ（ステップ S 2 1 3 0）。操作情報交換プログラム 2 0 1 は、自身のプロセス ID を受信待ちフラグに入れて「0」以外の値とし、排他制御を実現する（ステップ S 2 1

40)。

【0160】

排他制御セクションでは、HTTPサーバプログラム151からの応答データが通信制御プログラム202から転送されてくるのを待つ（ステップS2150）。尚、通信制御プログラム202が通信を行うコネクションは1本だけであり、また、HTTPサーバプログラム151がリクエストの到着順にレスポンスを返すので、受信データの順序が入れ替わって到着することはない。したがって、前記第2の実施形態における受信カウンタ処理は不要である。また、タイムアウト処理も通信制御プログラム202で行われるので、ここでは不要である。

【0161】

データを受信すると、前述したステップS816と同様、内容が操作情報を伝達するものであるか否かを判別する（ステップS2160）。操作情報であった場合、前述したステップS817と同様、操作データキューに追加する（ステップS2170）。要求カウンタをデクリメントし（ステップS2180）、受信待ちフラグを「0」に戻して排他制御セクションを抜け（ステップS2190）、ステップS2000に戻り、同様の処理を繰り返す。

【0162】

尚、操作情報交換プログラム201は、親プロセスである通信制御プログラム202からの割り込み処理等により、任意の時点で後処理を行った後に終了する。

【0163】

〔操作表示応答CGIプログラム211〕

操作表示応答CGIプログラム211の処理は、前記第2の実施形態における操作表示応答CGIプログラム752と基本的に同じである。ただし、複数のプログラムが並列動作するので、前記第2の実施形態で示したように、ステップS908では、要求カウンタの値がm未満である場合、操作イベントを待ち、m以上である場合、操作待ちループを抜けるようにする。さらに、ステップS909の操作イベント検出の前後では、操作情報交換プログラム201の状態変化待ちキューと同様、順に操作イベントを検出するようにする。

## 【 0 1 6 4 】

また、前述した通り、操作表示応答 C G I プログラム 2 1 1 からの応答の順序が保たれるので、送信カウンタ処理が不要である。したがって、送信カウンタ処理であるステップ S 9 1 1、S 9 1 2 の処理を削除し、ステップ S 9 1 3 において送信情報に送信カウンタ情報を付加する部分を除く。

## 【 0 1 6 5 】

このように、第 4 の実施形態では、H T T P の持続的コネクションとパイプライン処理機能を活用した、より少ない通信資源（通信コネクション）で（コネクションを開く）遅延のない通信が実現され、遠隔操作システムの通信性能が向上する。

## 【 0 1 6 6 】

また、通信を司るプログラムを分離独立させることにより、操作情報交換プログラムや操作表示応答 C G I プログラムを容易に持続的コネクションやパイプライン処理に対応させることが可能になる。尚、通信制御プログラムを独立させる形態は、前記第 1 の実施形態に示す状態通知プログラムと操作データ収集プログラムが分離された形態においても、容易に適用可能である。

## 【 0 1 6 7 】

この場合、状態通知プログラムおよび操作データ収集プログラムのそれぞれに対して通信制御プログラムを用意し、それぞれの通信制御プログラムにおいて通信を担わせる形態と、両プログラムに共通の通信制御プログラムを 1 つ用意し、両プログラムの通信を担わせる形態のどちらも可能である。さらに後者の場合、共通の通信制御プログラムが状態通知プログラムと操作データ収集プログラムのそれぞれに対して個別に通信コネクションを用意し、それぞれのデータを内部的に分離して通信することも可能である。このような各形態において、通信制御プログラム内で持続的コネクションとパイプライン処理機能を利用できることは明らかである。

## 【 0 1 6 8 】

以上が本発明の実施の形態の説明であるが、本発明は、これら実施の形態の構成に限られるものではなく、特許請求の範囲で示した機能、または実施の形態の

構成が持つ機能が達成できる構成であればどのようなものであっても適用可能である。

## 【0169】

例えば、上記各実施形態では、ファイアウォールの形態として通過パケット制限および中継プログラムによる形態を用いたが、本発明は、これに限らず、適用可能である。例えば、LANに接続される機器に対し、LAN内の専用のプライベートアドレスと呼ばれるアドレスを付与し、これをインターネットで用いられるグローバルなアドレスに変換するNAT (Network Address Translation) 機能をLANとインターネットの接続点に設け、LANから発せられた通信要求のみをアドレス変換して通信を可能にし、インターネットから発せられた通信要求に対してはアドレス変換をしないことで、ファイアウォール機能を実現している場合にもそのまま適用可能である。

## 【0170】

また、本発明は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記憶した記録媒体を用いて、システムあるいは装置にプログラムを供給することによって達成される場合にも適用できることはいうまでもない。この場合、記憶媒体から読み出されたプログラムコード自体が本発明の新規な機能を実現することになり、そのプログラム自体およびそのプログラムを記憶した記憶媒体は本発明を構成することになる。

## 【0171】

上記実施形態では、図3、図4、図6、図7、図9、図10、図12、図13、図14、図15、図16、図17、図19、図20、図21、図22のフローチャートに示すプログラムコードは記憶媒体であるROMに格納されている。プログラムコードを供給する記憶媒体としては、ROMに限らず、例えばフロッピー（登録商標）ディスク、ハードディスク、CD-ROM、CD-R、DVD、不揮発性のメモ리카ードなどを用いることができる。

## 【0172】

## 【発明の効果】

本発明によれば、ファイアウォールの内部に被操作装置を設置しても、インタ

ーネットに接続された操作端末からファイアウォールを越えて操作できる。

【図面の簡単な説明】

【図 1】

第 1 の実施形態における遠隔操作システムの構成を示す図である。

【図 2】

遠隔操作システムの動作シーケンスを示す図である。

【図 3】

状態通知プログラム 1 4 1 の処理手順を示すフローチャートである。

【図 4】

操作表示 C G I プログラム 1 5 2 の処理手順を示すフローチャートである。

【図 5】

情報交換テーブル 4 0 1 を示す図である。

【図 6】

操作データ収集プログラム 1 4 2 の処理手順を示すフローチャートである。

【図 7】

操作 C G I プログラム 1 5 3 の処理手順を示すフローチャートである。

【図 8】

第 2 の実施形態における動作シーケンスを示す図である。

【図 9】

操作情報交換プログラム 7 4 1 の処理手順を示すフローチャートである。

【図 1 0】

図 9 に基づく操作情報交換プログラム 7 4 1 の処理手順を示すフローチャートである。

【図 1 1】

カウンタテーブルを示す図である。

【図 1 2】

操作表示応答 C G I プログラム 7 5 2 の処理手順を示すフローチャートである。

【図 1 3】

図12につづく操作表示応答CGIプログラム752の処理手順を示すフローチャートである。

【図14】

第3の実施形態における状態通知プログラム10141の処理手順を示すフローチャートである。

【図15】

操作表示CGIプログラム10152の処理手順を示すフローチャートである。

【図16】

操作データ収集プログラム10142の処理手順を示すフローチャートである。

【図17】

操作CGIプログラム10153の処理手順を示すフローチャートである。

【図18】

第4の実施形態における遠隔操作システムの構成を示す図である。

【図19】

通信制御プログラム202の処理手順を示すフローチャートである。

【図20】

図19につづく通信制御プログラム202の処理手順を示すフローチャートである。

【図21】

操作情報交換プログラム201の処理手順を示すフローチャートである。

【図22】

図21につづく操作情報交換プログラム201の処理手順を示すフローチャートである。

【符号の説明】

- 100 インターネット
- 110 LAN
- 111 ファイアウォール

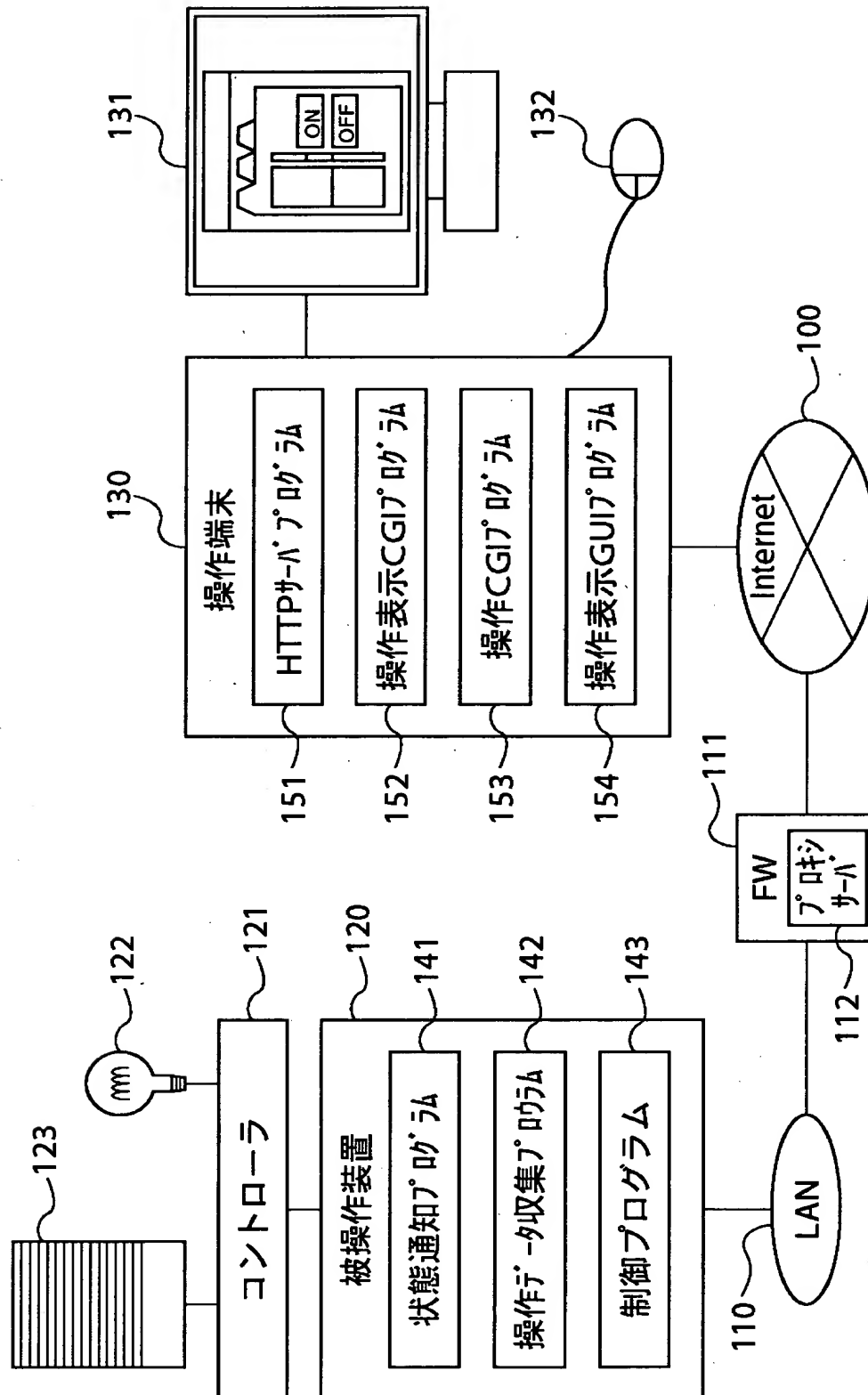


- 1 1 2 プロキシサーバ
- 1 2 0 被操作装置
- 1 2 1 コントローラ
- 1 2 2 照明
- 1 2 3 ブラインド
- 1 3 0 操作端末
- 1 4 1 状態通知プログラム
- 1 4 2 操作データ収集プログラム
- 1 4 3 制御プログラム
- 1 5 1 HTTPサーバプログラム
- 1 5 2 操作表示CGIプログラム
- 1 5 3 操作CGIプログラム
- 1 5 4 操作表示GUIプログラム
- 2 0 1 操作情報交換プログラム
- 2 1 1 操作表示応答CGIプログラム

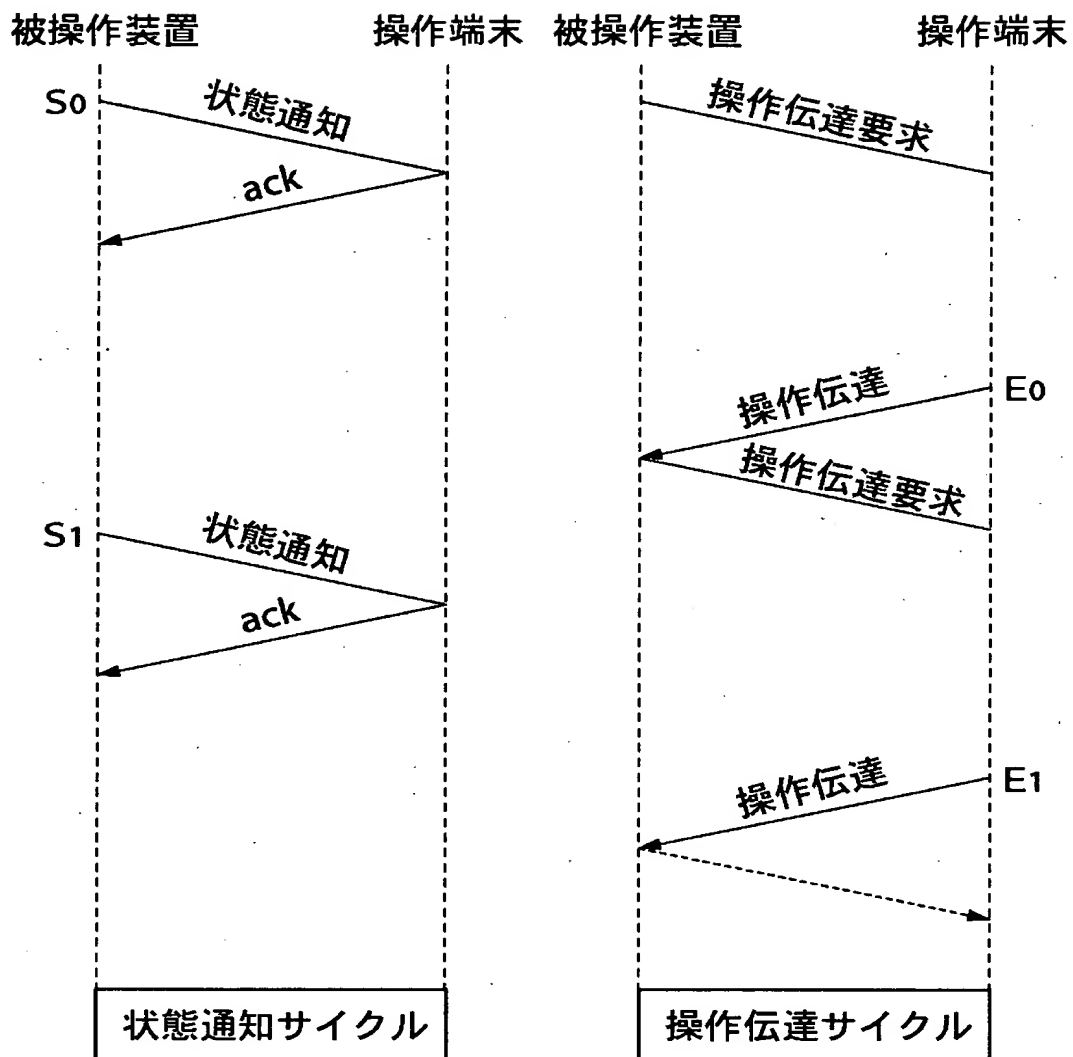
【書類名】

図面

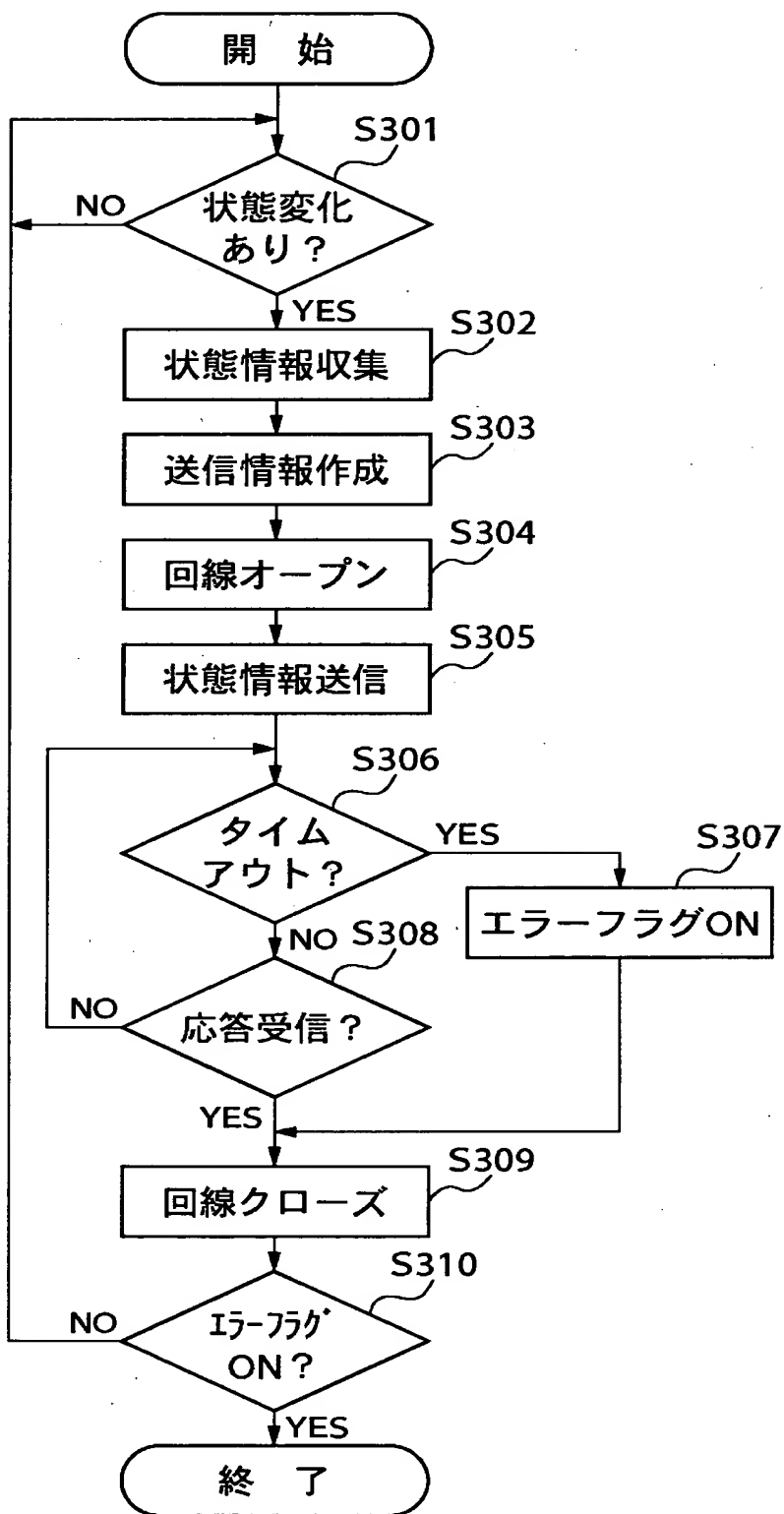
【図 1】



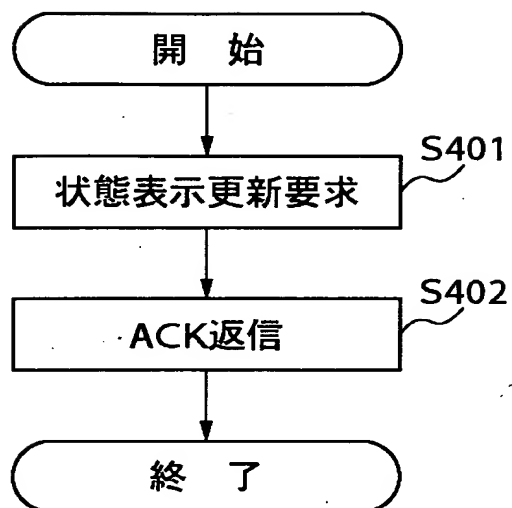
【図 2】



【図 3】



【図 4】

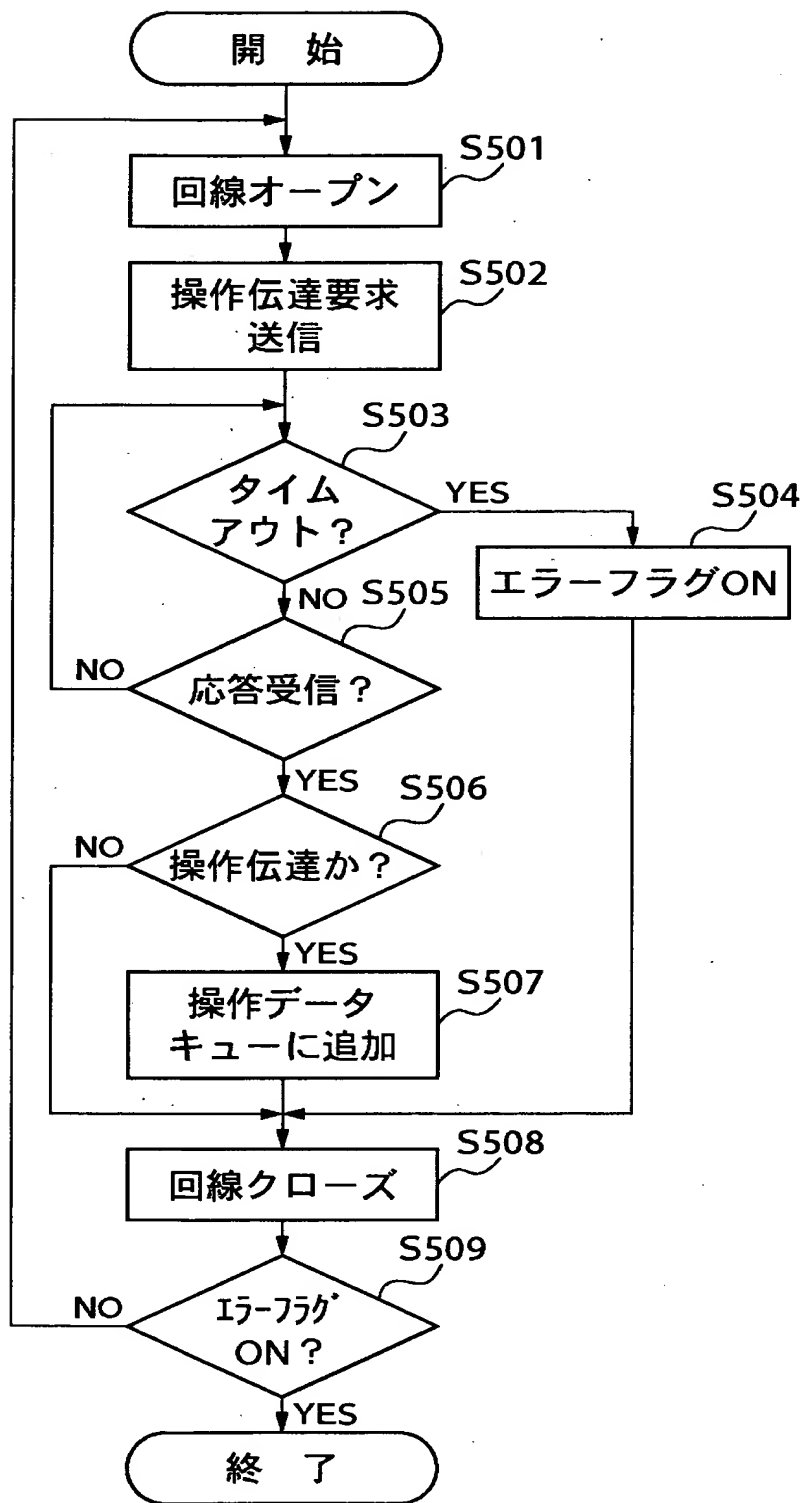


【図 5】

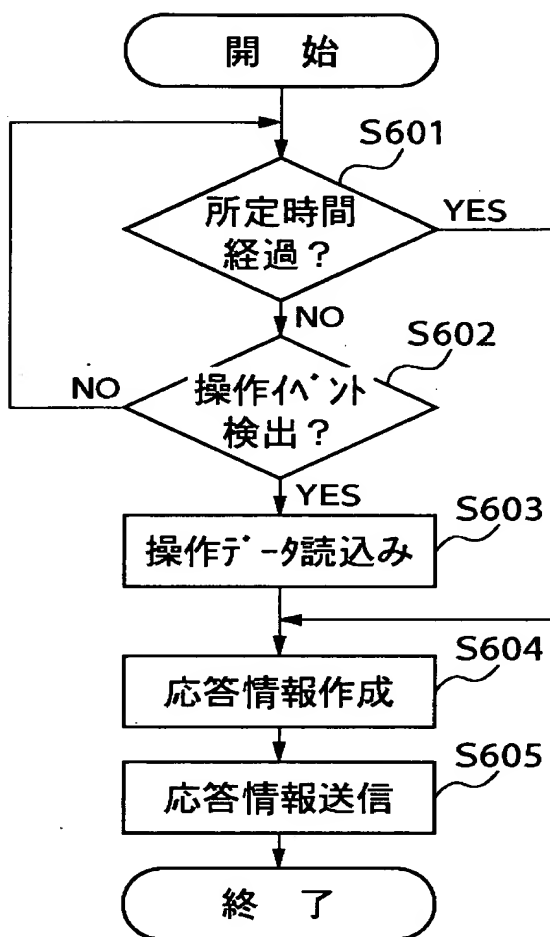
401

状態	light	OFF
	blind	70
要求	light	ON
	blind	80

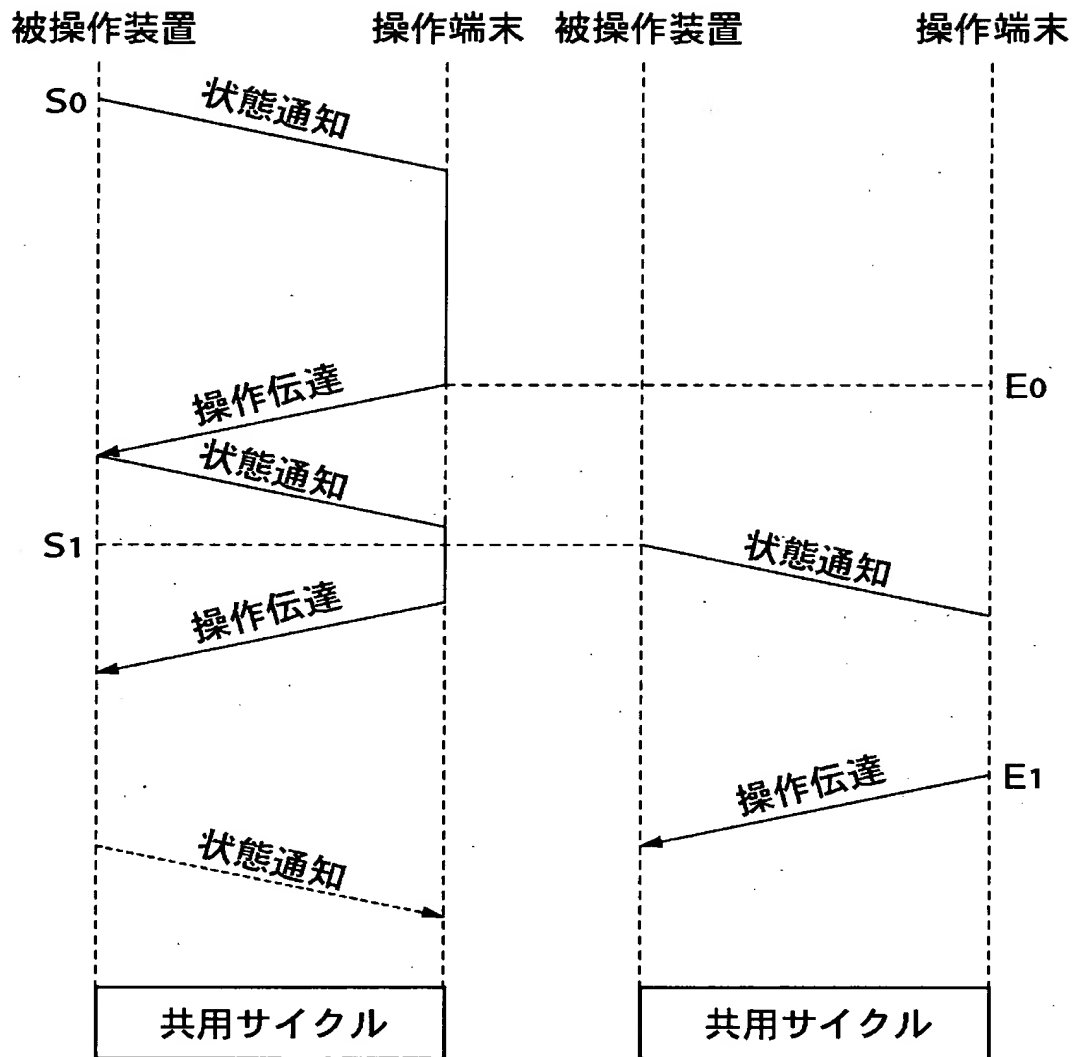
【図 6】



【図 7】

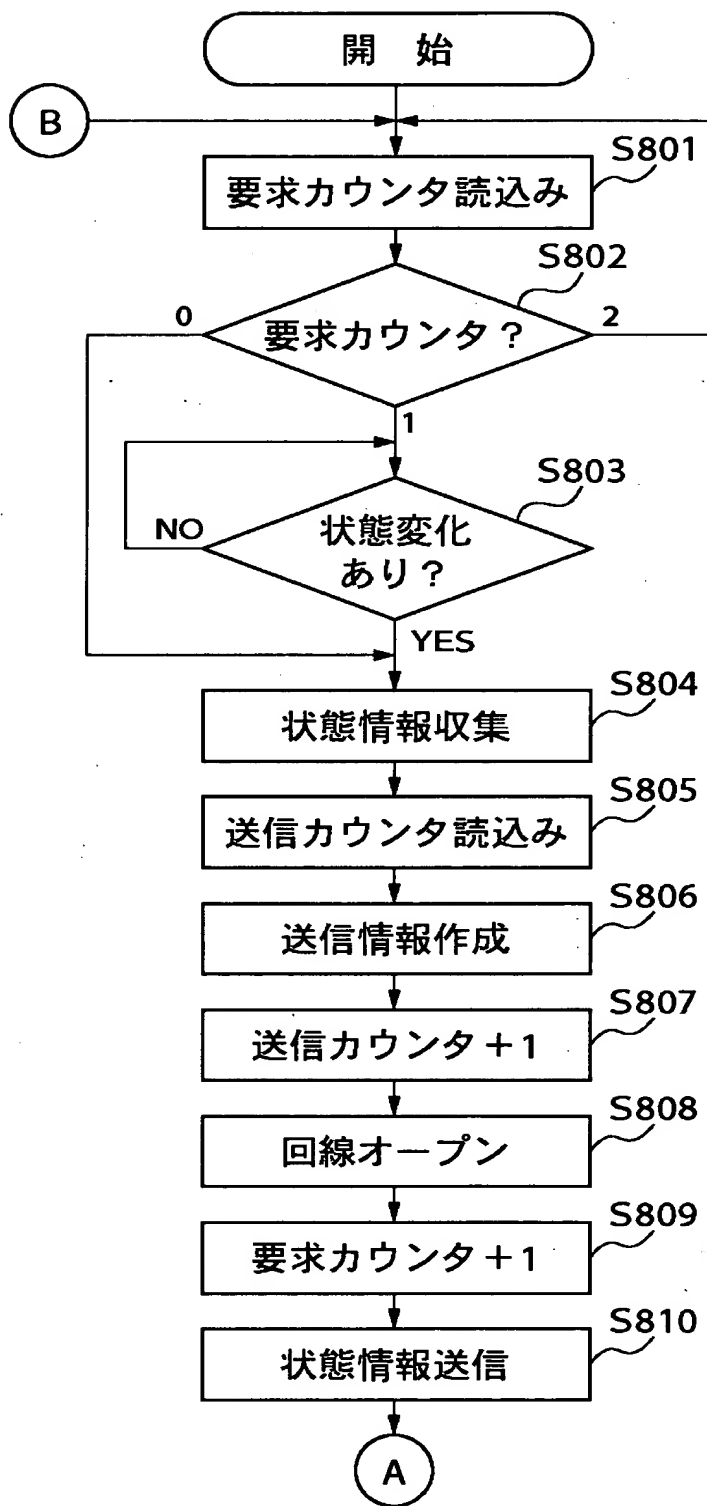


【図 8】

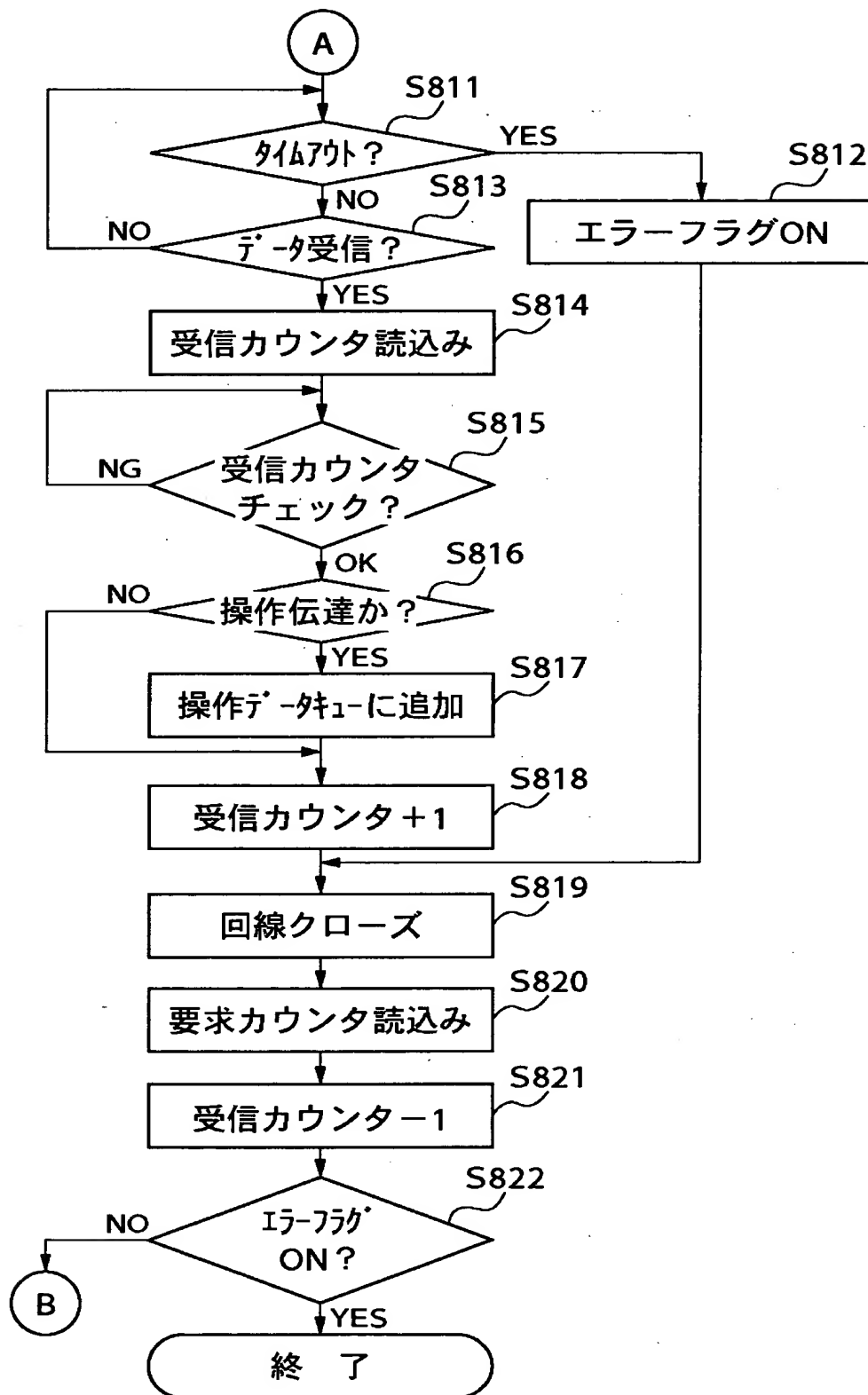




【図9】



【図10】

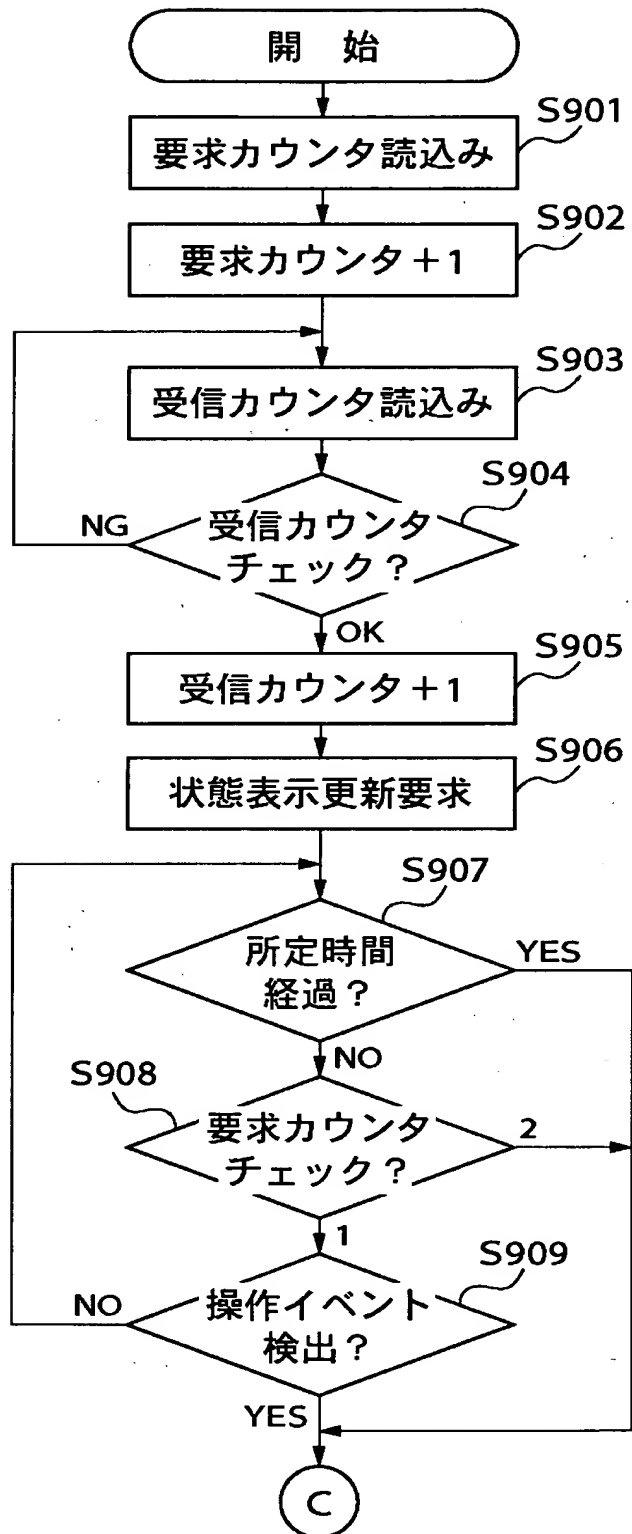


【図 1 1】

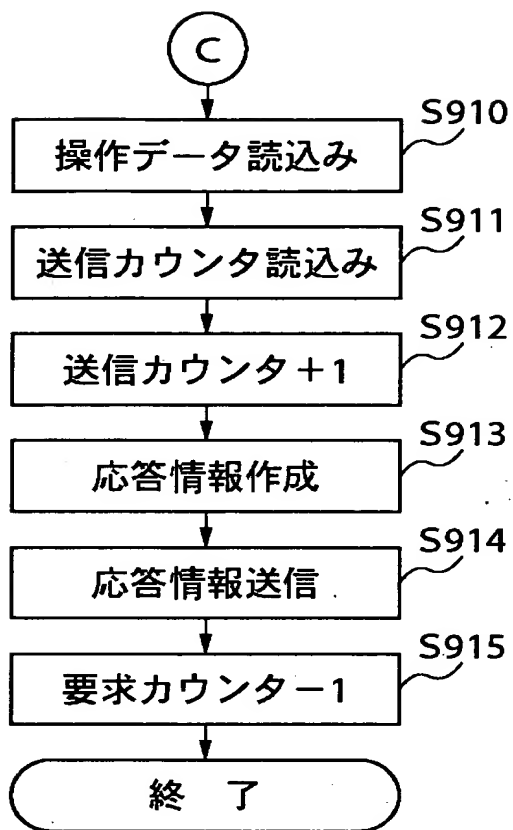
801

要求	0
送信	1
受信	0

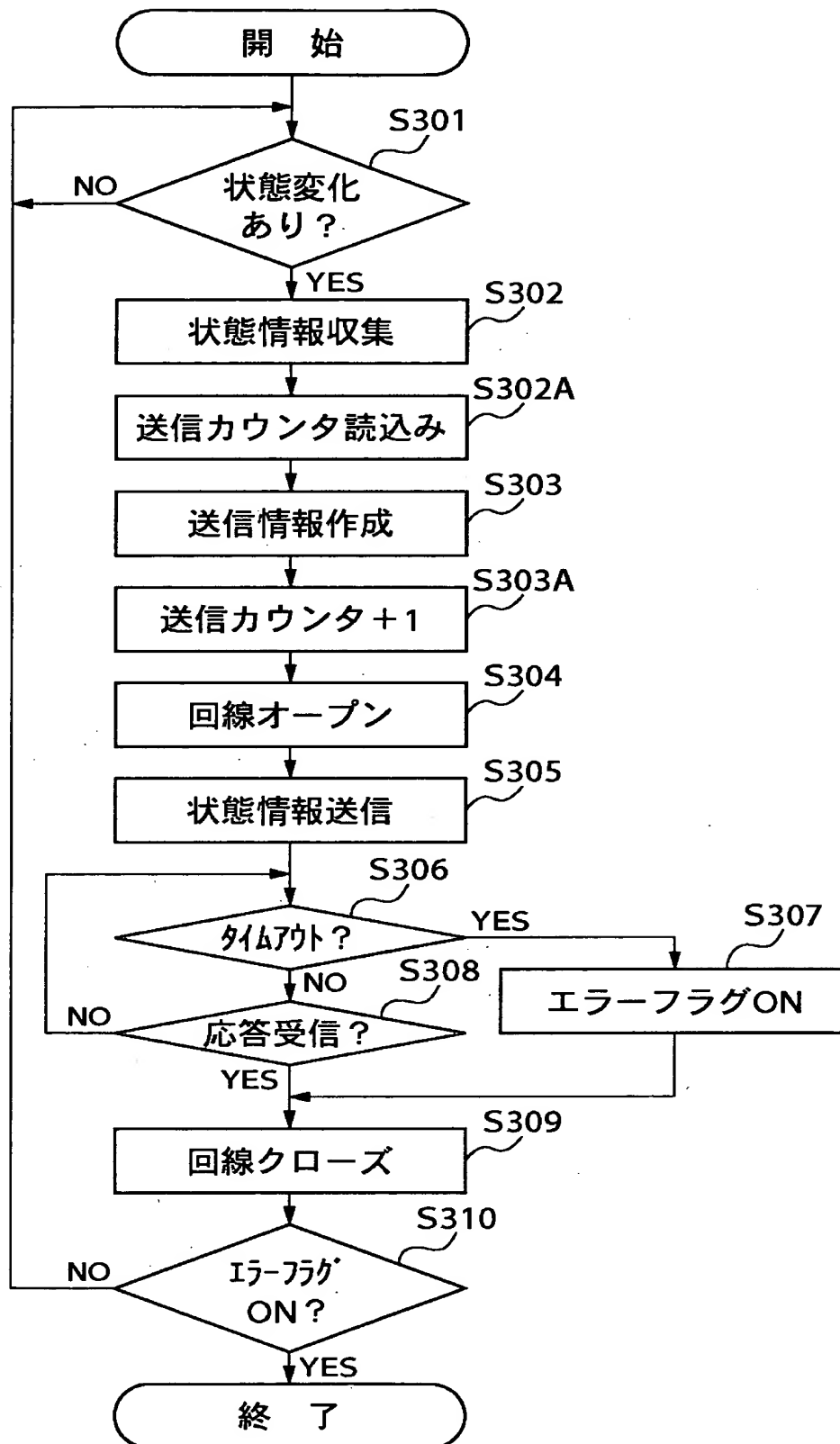
【図 1 2】



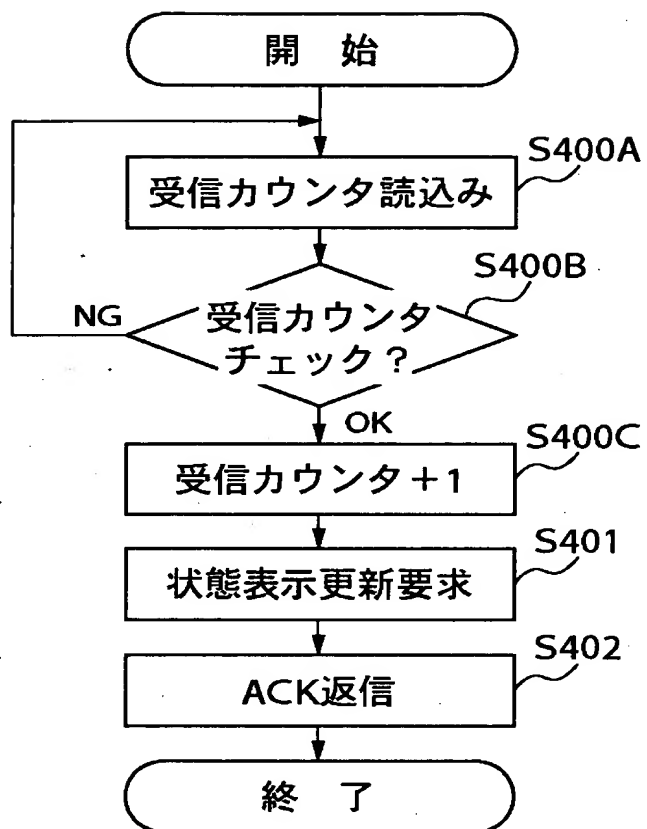
【図 1 3】



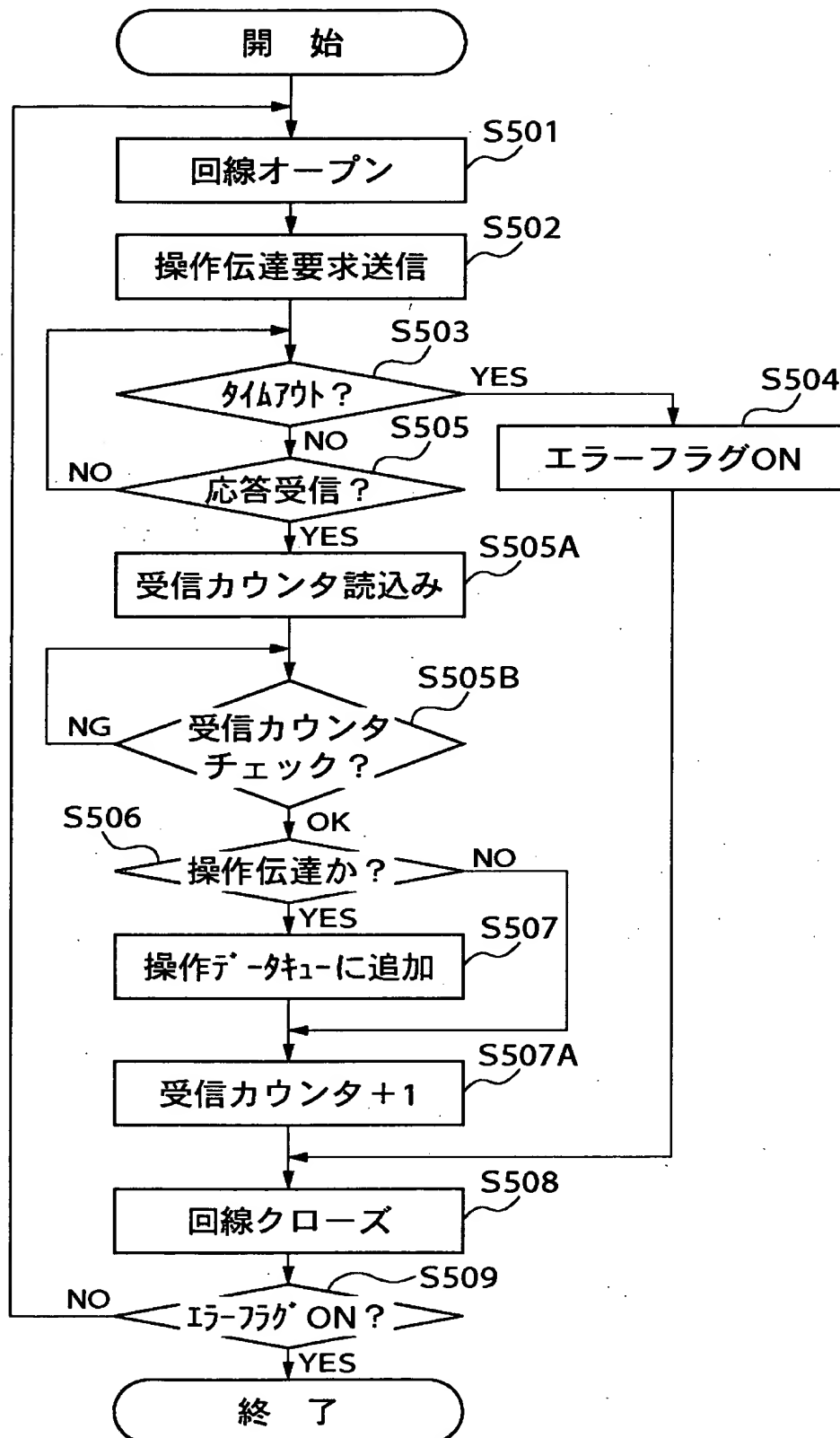
【図14】



【図15】

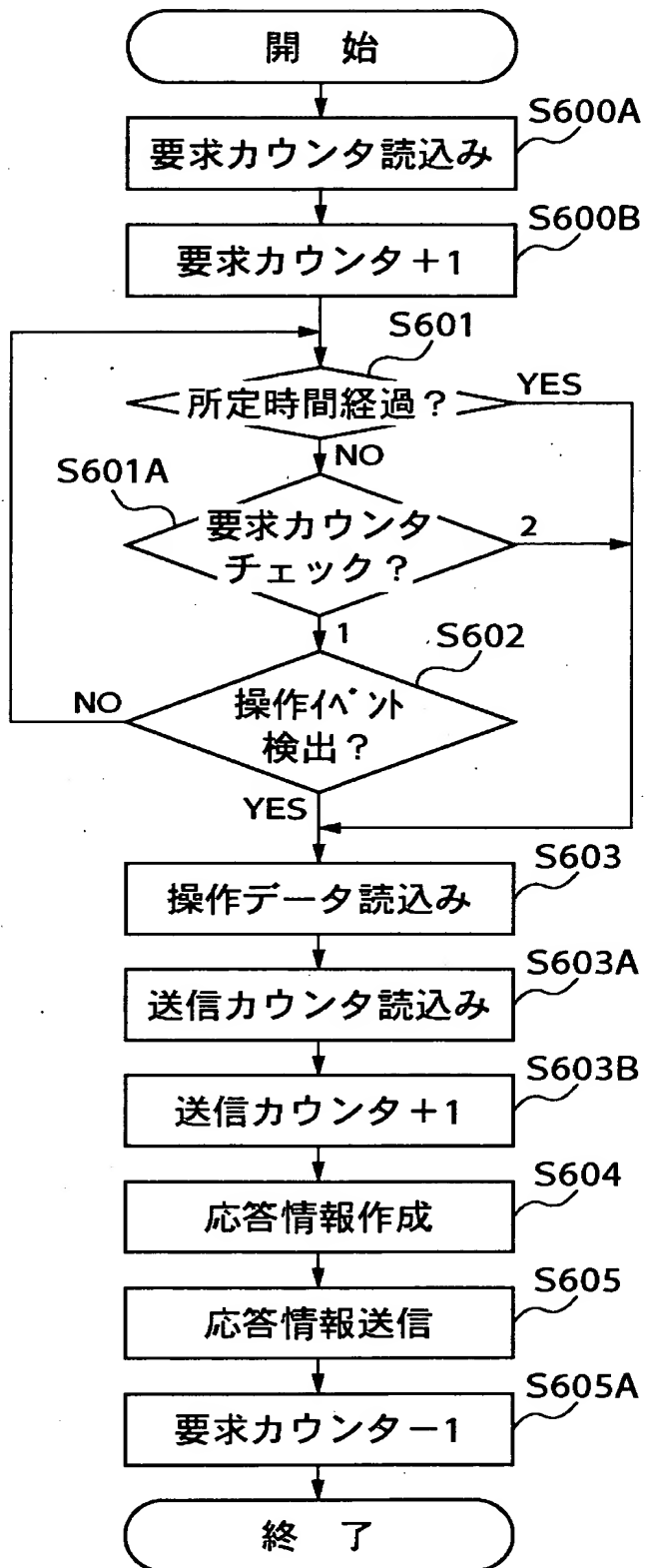


【図16】

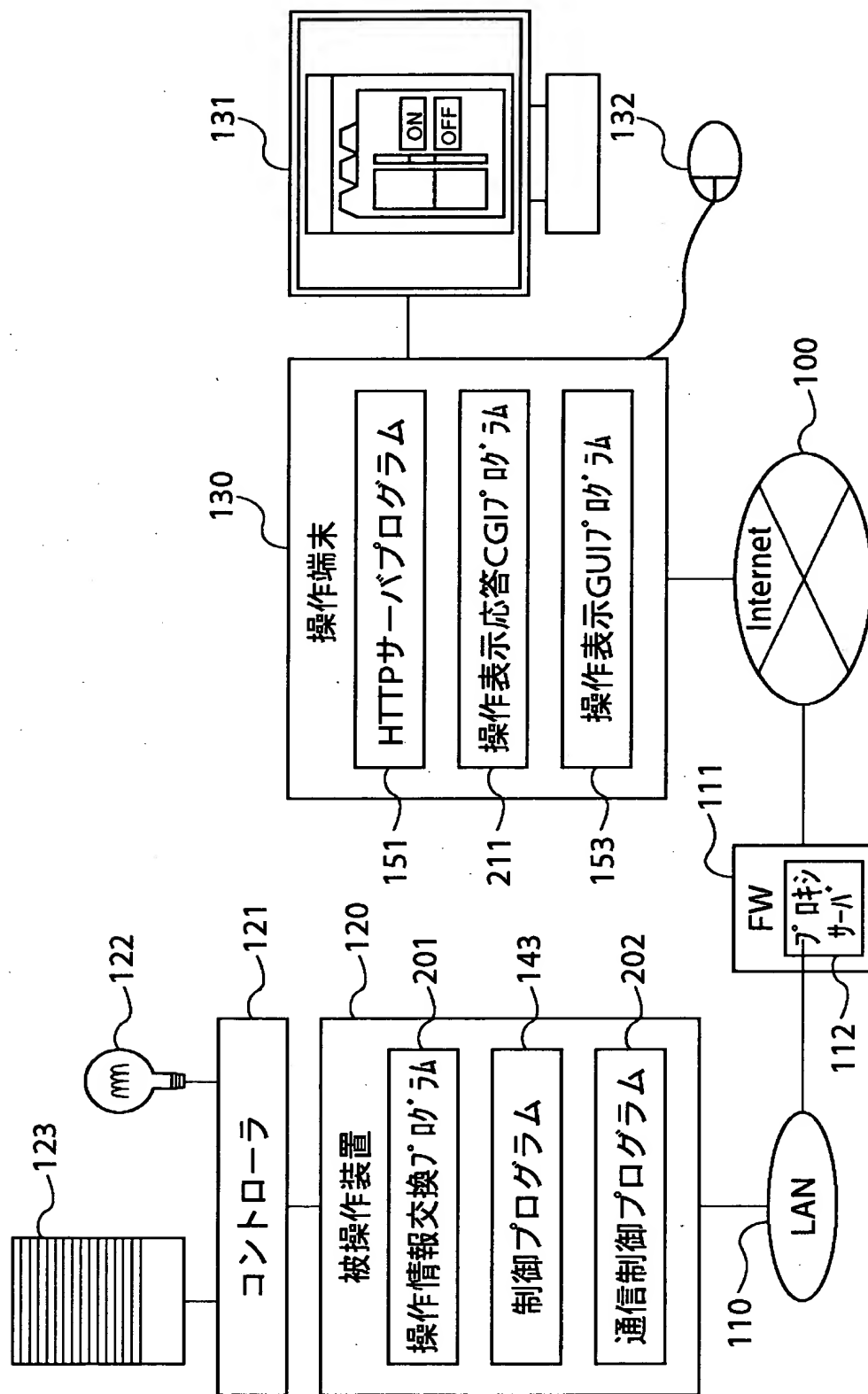




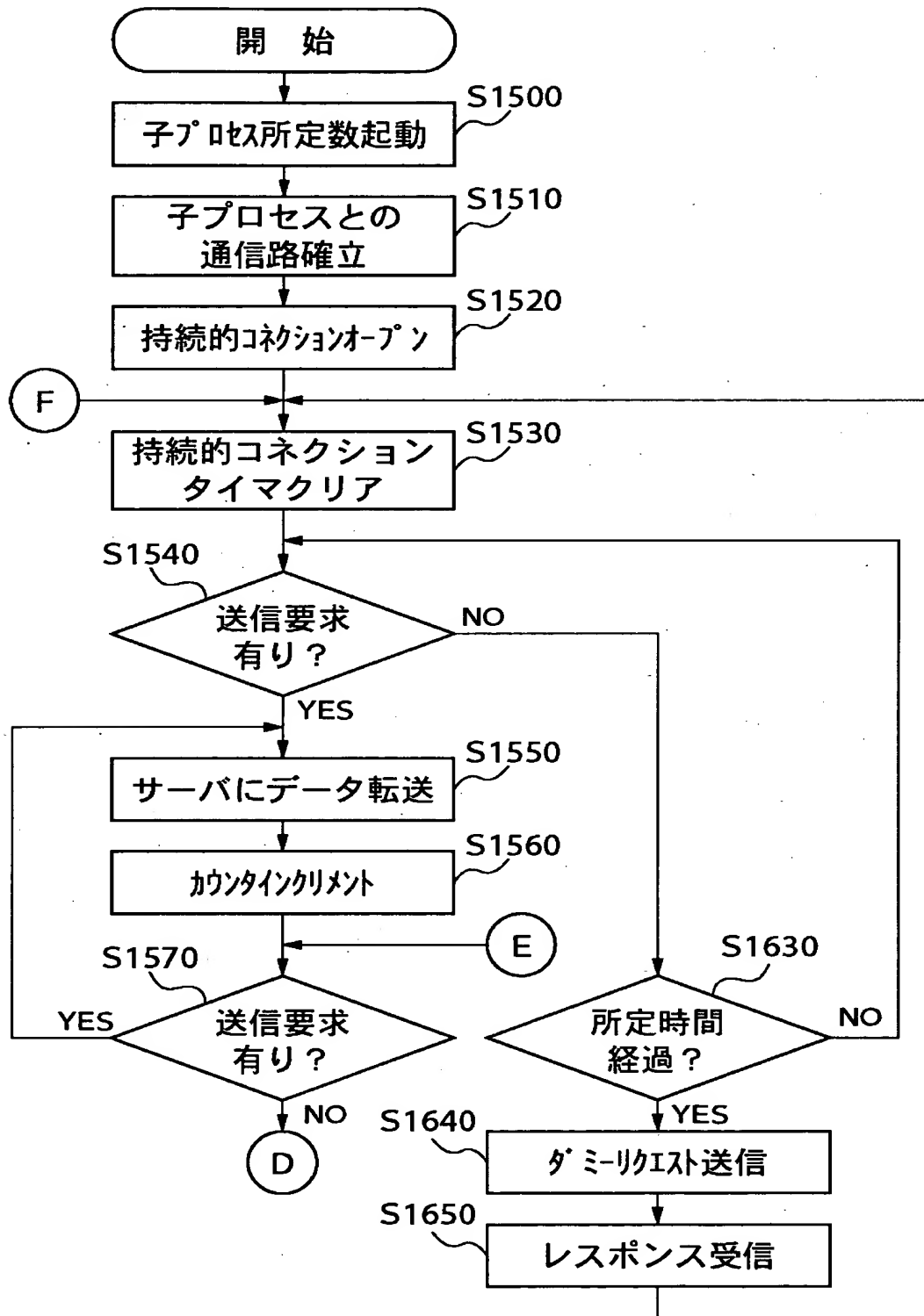
【図 17】



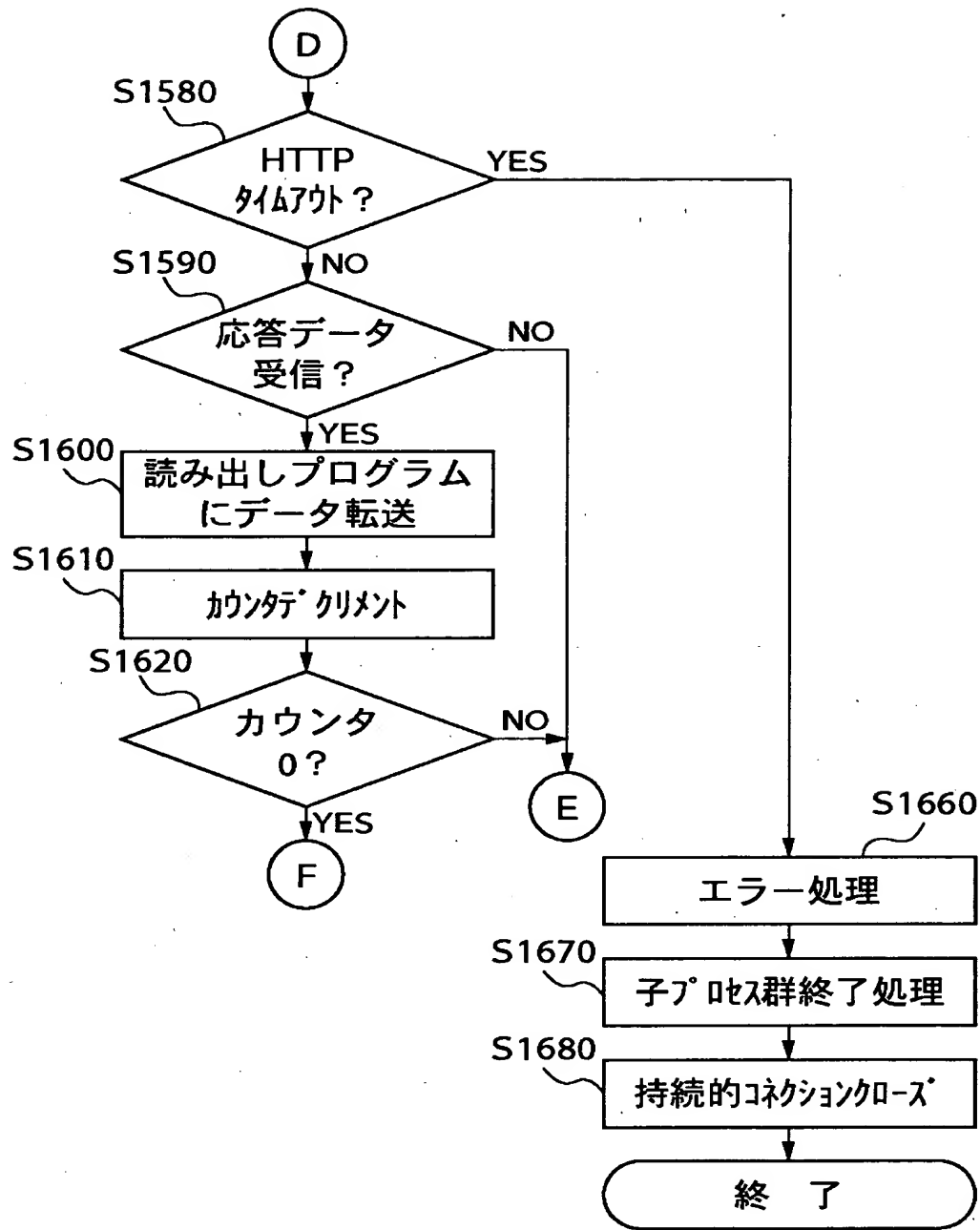
【図18】



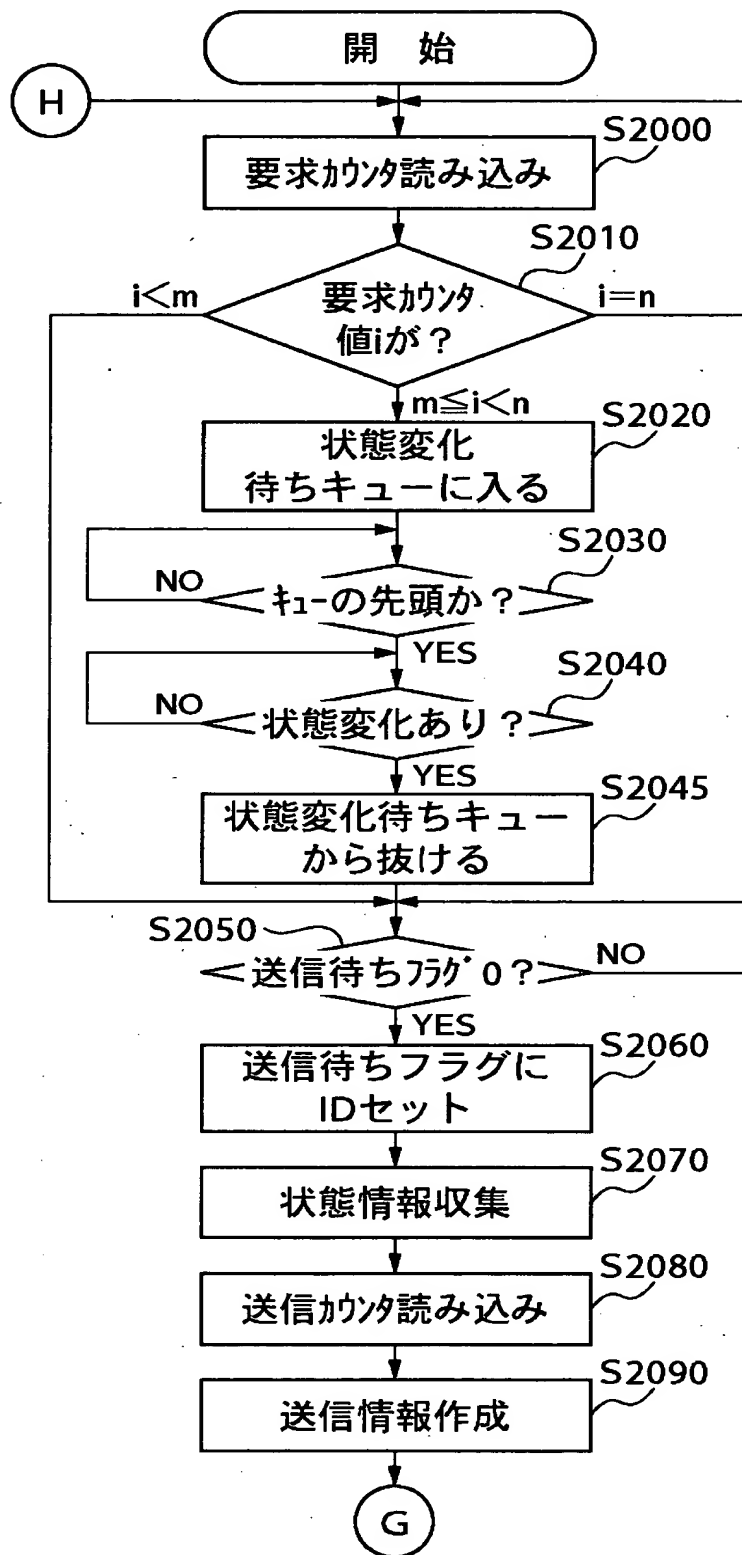
【図19】



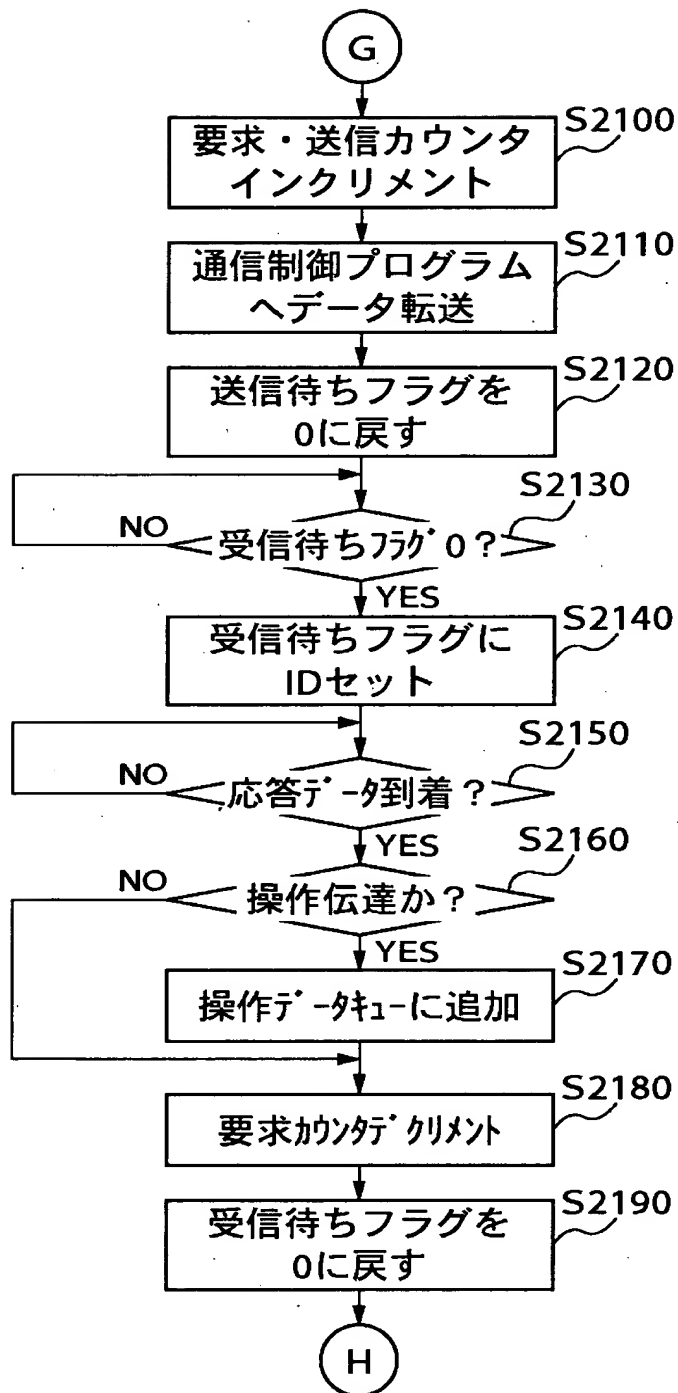
【図20】



【図 2 1】



【図 22】



【書類名】 要約書

【要約】

【課題】 ファイアウォールの内部に被操作装置を設置しても、操作端末からファイアウォールを越えて操作できる遠隔操作システムを提供する。

【解決手段】 ネットワークを介して被操作装置 1 2 0 および操作端末 1 3 0 が接続された遠隔操作システムでは、被操作装置 1 2 0 の状態情報および操作端末 1 3 0 に入力された操作情報を交換し、操作端末 1 3 0 から被操作装置 1 2 0 を操作する際、被操作装置 1 2 0 は、操作端末 1 3 0 への通信コネクションを発動し、被操作装置 1 2 0 の状態情報を通知すると、操作端末 1 3 0 は、通知された状態情報を表示するとともに、その受信応答を、即座に被操作装置 1 2 0 に返す。また、被操作装置 1 2 0 は、操作端末 1 3 0 への通信コネクションを発動し、操作情報を要求すると、操作端末 1 3 0 は、所定時間だけ操作の入力を待ってから、要求された操作情報を被操作装置 1 2 0 に返す。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日  
[変更理由] 新規登録  
住 所 東京都大田区下丸子3丁目30番2号  
氏 名 キヤノン株式会社